Copy ______

IDA PAPER P-2953

DOCUMENTATION OF THE FORCES MOBILIZATION MODEL (FORCEMOB) VERSIONS 3.1 AND 3.2

Volume I: General Description

Eleanor L. Schwartz
An-Jen Tai
Richard H. White
James S. Thomason, *Project Leader*

January 1996

Prepared for
Office of the Assistant Secretary of Defense
(Economic Security)

Approved for public release; distribution unlimited.

19960516 031

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DTIC QUALITY INSPECTED 1

IDA Log No. H 94-45791

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract DASW01 94 C 0054 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

IDA PAPER P-2953

DOCUMENTATION OF THE FORCES MOBILIZATION MODEL (FORCEMOB) VERSIONS 3.1 AND 3.2

Volume I: General Description

Eleanor L. Schwartz
An-Jen Tai
Richard H. White
James S. Thomason, *Project Leader*

January 1996

Approved for public release; distribution unlimited.

INSTITUTE FOR DEFENSE ANALYSES

Contract DASW01 94 C 0054
Task T-AO6-656

PREFACE

Since June of 1988 the Institute for Defense Analyses (IDA) has been assisting the Department of Defense in developing a systematic process to estimate U.S. stockpile requirements for strategic and critical materials. This paper documents the most recent versions of the Forces Mobilization Model (FORCEMOB), one of the quantitative computer models used in this process.

This paper was prepared in partial fulfillment of the task entitled "National Defense Stockpile Analyses." The task was performed for the Office of the Assistant Secretary of Defense (Economic Security).

While the authors are solely responsible for the substance of this paper, they would like to express their particular appreciation to the IDA reviewers, Dr. Lowell Bruce Anderson, Dr. Harry Gilman, and Mr. Stanley Horowitz, as well as to Dr. Paul Halpern of the Office of the Secretary of Defense, for their many constructive suggestions. Thanks are also due to Cori Bradford and Charlene Smith for preparing the manuscript, and to Shelley Smith for fine editorial assistance.

CONTENTS

FIGURES

TABLES

I. OVERVIEW

A. SCOPE AND OBJECTIVES

This paper documents Versions 3.1 and 3.2 of the Institute for Defense Analyses' Forces Mobilization Model (FORCEMOB). FORCEMOB is a model of the effect of an extraordinary military demand, such as a conflict or reconstitution, on the industrial base of the United States. Version 1.0 of FORCEMOB was documented in four volumes, which are now largely out of date because FORCEMOB has undergone extensive change, both in methodology and in computer program structure. Version 2 (comprising Versions 2.0, 2.1, and 2.2) and Version 3.0 were interim versions of FORCEMOB for which documentation was not updated. Differences between these interim versions and Versions 3.1 and 3.2 are noted where appropriate.

FORCEMOB Version 3.2 was developed after this documentation was nearly complete. Chapters I through IV of this volume are written as though Version 3.1 were the current version, but, except as noted, these chapters also apply to Version 3.2. Chapter V discusses the features specific to Version 3.2.

This documentation has several objectives:

- To present the FORCEMOB model in sufficient detail that an analyst or user can understand what it does and exactly what quantities are calculated
- To explain how to operate the FORCEMOB computer program
- To explain the FORCEMOB inputs in enough detail that an analyst or data preparer can understand precisely what kind of input data the model needs
- To describe the format of the FORCEMOB inputs in enough detail that a user or data preparer can make up input files in the correct format for FORCEMOB to read
- To present the differences between FORCEMOB Version 1.0 and the current version.

To these ends, this documentation is organized in two volumes, each with a number of different chapters.

Volume I focuses on FORCEMOB's methodology and describes all of its interactions. The description is fairly broad, but mathematical notation appears when necessary to clarify the methodology or the model's computations. The differences from Version 1.0 or interim versions are noted.

Volume II, the *Data Preparation Guide*, describes in detail the structure of the model's input data files. It provides definitions of every input variable to the model, and describes exactly how the input files are formatted.

The balance of this chapter provides a basic understanding of FORCEMOB. Section B tells how FORCEMOB is structured and what it does. Section C summarizes some underlying concepts. Section D presents, in list form, the major differences between FORCEMOB Versions 1.0 and 3.1. Section E outlines the structure of the balance of Volume I.

B. WHAT IS FORCEMOB?

FORCEMOB is one component of the Joint Industrial Mobilization Planning Process (JIMPP), an analytic process that links warfighting needs with the capabilities of the industrial base [1, 2]. FORCEMOB is designed to determine and examine the effect of an extraordinary military demand, such as a conflict, on the U.S. industrial base. The model can consider demands from all Services and can take into account the capacities of a set of industries that span the economy, including the "lower tier" of industries that have an indirect, yet possibly significant, effect on defense production. In addition to the extraordinary military demand, it also considers the effects of peacetime military demand, civilian demand, and investment demand, and thus provides a comprehensive, broad-brush picture of the effects on the whole economy of the extraordinary military demand.

FORCEMOB comprises two modules, the Requirements module (abbreviated REQMOD) and the Industry-level module (ILM). The Requirements module computes a time-phased schedule of weapon requirements associated with a conflict (or other extraordinary military demand) and translates these weapon demands into time-phased demands on industry. The Industry-level module then does the following:

1) Takes the conflict-induced industry demands
2) Adds in civilian and base military demand on industry
3) Determines the industry supply capacity (domestic production plus imports)

4) Compares demand against supply and determines whether or not any shortfalls occur (early shortfalls cannot be offset by later surpluses)

5) Models the process of investment to redress shortfalls and computes investment demand

A major use of FORCEMOB has been to evaluate requirements for the National Defense Stockpile [3]. This evaluation is a three-step process that integrates forecasts and policy judgments from throughout government and industry. The three steps are as follows:

1. Determine requirements for goods and services over an appropriate scenario. Consider civilian demand, normal defense demand, extraordinary defense demand, and investment demand.
2. Determine the materials required to produce those goods and services.
3. Compare the requirements for materials with the corresponding supplies. Shortfalls become candidates for the National Defense Stockpile.

FORCEMOB—both the Requirements and Industry-level modules—is used in step 1. Steps 2 and 3 are addressed using another module of JIMPP, the Stockpile Sizing Module [4]. FORCEMOB has also been used in several other studies of mobilization and the industrial base, most recently in a study in support of the Naval Logistics 1994 Wargame [2].

C. FORCEMOB BASICS

This section presents a brief summary of what FORCEMOB does and discusses some concepts that undergird the modeling process in FORCEMOB.

1. Basic Constructs

FORCEMOB assesses the effect of an extraordinary military demand on the industrial base. It assesses the ability of the industrial base to build the extra weapons and other military items necessary to meet this demand within a specified time period. It computes shortfalls in supply, if they exist, and examines options to redress these shortfalls, either by more fully utilizing existing plant capacity or by investing in new plant and equipment.

Among the fundamental inputs to FORCEMOB are:

1. A list of weapon systems, or "Major End Items" (MEIs), that can be considered[1]

2. A list of industry sectors that span the U.S. economy (in this documentation, the terms "industry," "sector," and "industry sector" are used synonymously)

Most FORCEMOB results are tracked at the level of detail of these lists, but not finer.[2] The lists can be made as coarse or as fine-grained as the user desires, subject to availability of accurate data. In the analyses performed with FORCEMOB so far, the industry sectors generally correspond to the list of 4-digit Standard Industrial Classification sector codes developed by the Office of Management and Budget [6], with some amount of condensation and aggregation.

FORCEMOB explicitly considers the impact of time. The user specifies an overall scenario period, and for each month within this time span, supply, demand, and shortfall values for that month are input or computed.[3] This time-phased framework allows FORCEMOB to consider such things as lead times for production of MEIs, lead times for capacity expansion or investment in new facilities, early shortfalls which cannot be redressed by later increases in supply, and the like. Within the scenario period, a conflict start date is specified; the time between the scenario start date and the conflict start date models the concept of a mobilization period. If a multitheater conflict is being modeled, each theater can have its own separate conflict start date and conflict length. The time span between the end of the conflict and the end of the scenario can be used to model a return to peacetime conditions. Chapter IV, section B.1, discusses FORCEMOB's time modeling in more detail, from the standpoint of the computer program.

Many of the FORCEMOB data and results are expressed in monetary terms, e.g., thousands of dollars or millions of dollars. The year in which these dollar values are

---

1 An end item is the final combination of end products, component parts, and materials which is ready for its intended use, e.g., a ship, tank, mobile machine shop, or aircraft. A Major End Item is one of a limited number which, for reason of military urgency, criticality, or resource requirements, is determined by the Department of Defense to be vital to the national interest.

2 As discussed in Chapter II, section A.1, below, the Requirements module can operate on data about weapons and consumables; these data might be specified at a more detailed level than the MEIs. Major results of the Requirements module are expressed at the MEI level of aggregation. Some of the output reports show the finer level of detail.

3 The scenario period can be up to 15 years long. No methodological changes to FORCEMOB would be necessary to accommodate a longer scenario, but some changes to the computer program would be required.

expressed must be the same for all data. FORCEMOB performs no automatic adjustment for inflation.[4] A "dollar year" (e.g., 1987) is input to FORCEMOB in the Element data base file (see Volume II), and all the monetary input data are assumed to be expressed in dollars of that year. In preparing the input files, the user must check that this condition holds. Monetary FORCEMOB results are also expressed in dollars of that year.

2. Steps of the Model

The steps of FORCEMOB are as follows:

1. Based on inputs, determine a time phased stream of force units (of various types, possibly from several different Services) participating in a (possibly multitheater) conflict. Determine the weapons and consumables required to support this conflict for a user-specified period of time.
2. From the conflict specifications, determine the dollar amounts of the Major End Items demanded, by month and MEI. (An option exists by which these amounts can be input directly.)
3. Translate the MEI demand into demand on the industry sectors.
4. To this "conflict military demand" on industry, add base (peacetime, normal) military and civilian demands to determine total demand, for each month and industry.
5. Determine the domestic supply available in each industry sector, by month. Modify this supply to take into account expanded utilization of capacity, considering dual use as appropriate. Add in net imports.
6. Compare the supply with the demand; compute and report the shortfalls, if any. Model the process of investment to redress the shortfalls, and report the results. (Investment demand created by the investment process becomes an additional source of demand. The additional supply created by investment is available to help offset all forms of demand.)

The Requirements module encompasses steps 1, 2, and 3; the Industry-level module, steps 4, 5, and 6. The two modules communicate via a "Conflict Military Requirements file," which specifies, for each month and industry, the demands on industry that are attributable to the conflict. A given run of FORCEMOB can exercise either the Requirements module, the Industry-level module, or both, as the user requests. If only

---

4 A data file preprocessor program for FORCEMOB Version 1.0 did perform some inflation adjustments, but this preprocessor is not used with Version 3. It could, however, be reactivated at some point.

the Industry-level module is exercised, however, the user must input to it a Conflict Military Requirements file of some sort. This file will generally have been generated by a previous run of the Requirements module (but it can be prepared directly by the user; see Volume II).

3. Inputs and Outputs

a. Inputs

FORCEMOB requires a considerable amount of input data, which is organized into a number of different data files. The Requirements module accepts a large set of data that is used to compute the weapons and consumables associated with a conflict scenario. The translation of weapon requirements into industry requirements is accomplished via a large data file that specifies the industry demands associated with each type of Major End Item. The Industry-level module requires data on peacetime military demand, civilian demand, industry plant capacity, and investment characteristics. Volume II of this paper constitutes a comprehensive description of all the input data associated with the FORCEMOB model. (The kinds of input data that Version 1.0 used, as described in [1] and [7], are still used in Version 3.1, but the format of the associated data files and the precise nature of the data have changed somewhat.)

Among the inputs, the "Control Inputs" data file deserves special mention. Each run of FORCEMOB is associated with exactly one Control Inputs file. It lists the names of the other input data files to be used in the run. It also specifies—

- scenario dates
- sensitivity parameters that can be used to alter the file data
- parameters that allow the user to select certain options
- a list of output reports to be generated in the run

The user can put informative comments about the run at the bottom of the Control Inputs file. For complete information on the data and format of the file, see Volume II, Chapter II.

b. Outputs

At the user's request, FORCEMOB can generate a number of different output reports. These display various results computed by the model. Further information on these reports appears in Chapter IV of this volume and Chapter II of Volume II. Each run

of FORCEMOB generates a "history file," which shows (a summary of) the inputs to the run and certain summary output measures. If only the Requirements module is being exercised in a run, then a Conflict Military Requirements file is generated automatically.[5] This file can then be input to subsequent runs of the Industry-level module.

4. Appropriate Questions for FORCEMOB

As indicated earlier, FORCEMOB tracks its results at the level of industry sectors and MEIs, on a time period by time period basis. An analyst can use FORCEMOB to investigate questions such as the following:

- What shortfalls, in what industries, would be caused by the force structure demands associated with a given conflict?
- What combination of supply expansion and/or investment might ameliorate these shortfalls? To what extent?
- Could shortfalls be alleviated by imposing civilian austerity? By how much?
- How much more of the conflict demand could be met if the mobilization period were lengthened (to allow for more capacity expansion, investment, and production)?
- Could the use of different kinds of weapons divert demand from heavily used industries?

We stress that FORCEMOB acts as a "rough filter" and identifies areas that are exceptionally problematic. Thus a report by FORCEMOB that there is a large shortfall in the shipbuilding industry underscores the need for examination of the shipbuilding industry at a more detailed level. However, even if FORCEMOB reports that no shortfall exists, bottlenecks, delays, or supply/demand mismatches might become apparent when industries are examined at a finer level of detail.

D. SUMMARY OF DIFFERENCES FROM FORCEMOB VERSION 1.0

This section presents, in list form, the major differences between FORCEMOB Version 3.1 and Version 1.0. This section is intended primarily for those readers who are familiar with Version 1.0. The differences can be roughly grouped into three categories:

- Differences in methodology

---

5 If the run is exercising both modules, then the Conflict Military Requirements file can be generated at the user's request.

- Differences in computer program structure
- Differences in input data structure

Most of the methodology and computer program changes apply to Version 2.2 and 3.0 as well as Version 3.1. The input data file structure is similar for Versions 3.1 and 3.0. Data for Version 2.2 have some characteristics in common with Version 1.0, others with Version 3.

1. Differences in Methodology

Supply-side modeling. The supply-side methodology is considerably different from that of Version 1.0. There are major differences in the modeling of supply expansion and investment. The supply-side modeling of Version 1.0 is explained in [5] and [8]. A major objective of the current document is to explain the supply-side modeling of Version 3.1.

Dual use. Previous FORCEMOB versions assumed complete civilian/military fungibility in productive capacity: capacity that was used to make civilian goods could also make military goods. Version 3.1, by means of an optional input file, allows limited (or zero) dual use capability.

Labor. Version 1.0 did have some treatment of labor requirements and constraints. Version 3.1 currently does not work labor constraints into the new supply-side methodology. Partially, this is due to lack of available data. Modeling of labor constraints will probably be added to future FORCEMOB versions.

Production process. A minor change was made in the production process methodology, which translates demand for weapon systems into demand on industry. For each combination of industry and weapon system, the computation of the total amount of industry demand per amount of weapon system demand is the same as before, but now, this demand is assumed to be spread out evenly over the production lead time of the weapon system. Version 1.0 allowed a general spread pattern, which was specified in an exceedingly large and cumbersome-to-use input file. This slight methodological restriction has greatly reduced FORCEMOB's running time and has led to far easier file management.

Module structure. While FORCEMOB always has comprised two modules, the Requirements module and the Industry-level module, Version 3.1 treats them separately far more than Version 1.0 did. Communication between the modules is now accomplished via one file, the Conflict Military Requirements file. A given run of FORCEMOB Version 3.1 can run either or both modules, as the user requests, and many ILM runs can be made using the results of one Requirements

module run. Version 1.0 usually ran both modules. (The Requirements module itself is virtually the same as in Version 1.0.)

2. Computer Program Differences

Version 1.0 was an interactive program. Version 3.1 can be run in batch mode. Data that used to be entered on-screen [1] now is contained in the Control Inputs data file (see Volume II).

Version 1.0 used the VAX Forms Management System (FMS), a software package that could operate on VAX computers only. FMS provided the vehicle for interactive user input, and many FMS commands and function calls appeared in the FORCEMOB computer code. Version 3.1 is written in FORTRAN that is fairly close to the ANSI standard and should be portable to many different machines and FORTRAN compilers (see Chapter IV). A Graphical User Interface program, which runs on a PC under Microsoft Windows, has been developed to integrate and facilitate FORCEMOB data preparation, program execution, and analysis, but the interface is a separate program from FORCEMOB itself.

Version 1.0 allowed the user to make changes to certain input data in the middle of a run; the program would then recompute results based on the new data values. In Version 3, one run of FORCEMOB uses one set of input data and produces one set of outputs. (One execution of the FORCEMOB computer program can process more than one run of the FORCEMOB model.)

Version 1.0 had a number of different output reports and graphs that could be displayed on screen during the course of a FORCEMOB run. This information now appears in reports that are written to output files. These files can be printed, viewed on screen with a text editor, or pulled into a spreadsheet for analysis—but after the FORCEMOB program has stopped executing.

Several long, large subroutines in the Version 1.0 computer code have been broken up into smaller subroutines.

3. Differences in Input Data

Between FORCEMOB 3.1 and previous versions, there are a large number of differences in the structure and contents of many of the input data files. These differences are listed in Volume II, Chapter I and discussed throughout that volume.

E. STRUCTURE OF VOLUME I

As we proceed in Volume I, Chapter II elaborates on the material of Chapter I, section C, delineating all of the interactions and calculations of FORCEMOB in the order

that they are modeled within a run of the FORCEMOB computer program. First, the Requirements module is described, and then the Industry-level module.

The major methodological change to FORCEMOB from Version 1.0 is the way investment in new capacity is modeled. Chapter III describes the Version 3.1 investment algorithm in detail.

Chapter IV is devoted to the FORCEMOB computer program (as opposed to the FORCEMOB model). Unlike the Version 1.0 *Programmers Guide* [5], Chapter IV is not intended to be a detailed guide to the computer code. Instead, it provides an overview of the code and explains how to run the program. An annotated list of subroutines is included. Written in FORTRAN, FORCEMOB now runs on a VAX or PC. Chapter IV provides some information on converting the program to other machines.

Chapter V discusses FORCEMOB Version 3.2, describing how it differs from Version 3.1.

II. INTERACTIONS AND CALCULATIONS

This chapter delineates the interactions and calculations of the FORCEMOB model. The order of discussion corresponds to the order in which the interactions are modeled during a typical FORCEMOB run. The reader is assumed to be familiar with the overview of FORCEMOB presented in Chapter I.

The discussion includes mathematical notation when necessary but avoids the use of computer program variable, array, and subroutine names. Volume II contains definitions of every input variable in the computer program, along with some explanatory notes. This information can augment and supplement the exposition of the current chapter. Specific sections of Volume II are referenced as appropriate. Names of input data files are mentioned at times; see the corresponding sections of Volume II for more information about these files.

A. REQUIREMENTS MODULE INTERACTIONS

The Requirements module of FORCEMOB determines the demand for Major End Items that arises from a conflict or other extraordinary military demand. It then reduces these MEI demands by available inventory. Finally, it converts the net demands for MEIs into demands on industry. Demands are time-phased: for each month of the scenario period, values are computed that represent the demands in that month.

For the first step of this process—determination of the MEI demand—the model provides two alternative methods, discussed in sections A.1 and A.2, respectively. In the first method, the MEI requirements are computed from a conflict scenario. In the second method, the MEI requirements are input. Sections A.3 and A.4 discuss the additional steps of the Requirements module.

The Requirements module has changed very little from Version 1.0, and the description of the Requirements module in the Version 1.0 documentation ([5], [8]) might provide additional information to the reader. Additional references on the Requirements module include [3, Appendix B] and [9].

1. Conflict Specification and Computation of Major End Item Demand

This section summarizes the way FORCEMOB models the conflict situation that is used to determine the extraordinary military demand for Major End Items. Note that this is a mobilization planning scenario, not a two-sided conflict simulation. The point is not to compute the losses that would actually be suffered but to determine the number of Major End Items to plan to have on hand at a given time. Enemy forces are not explicitly modeled.

This portion of FORCEMOB uses a large amount of input data. Most of these data are contained in the Force Structure database file. Item names are specified in the Element database file, and item costs in the Cost database file. All these files, and the data therein, are discussed in detail in Volume II.

In this section, subsections 1.a through 1.c describe the constructs that the requirements calculation deals with. Then, subsection 1.d puts all this information together and describes how the requirements amounts are computed. (In the description of this portion of the Requirements module, the terms "demand" and "requirements" are used synonymously.)

a. Conflict-Related Constructs

Theaters. FORCEMOB can model a multitheater conflict, involving up to four theaters. The Force Structure input file contains data for each of a possible number of theaters, up to a maximum of four.[1] On the Control Inputs file, the user can specify that only certain of these theaters be played (i.e., simulated) in the run. Each played theater is given (on the Control Inputs file) a conflict length (in months) and a conflict start date. These lengths and dates can be different for different theaters. In addition to the usual interpretation of a multitheater conflict, different theaters can be used to simulate different portions of a conflict, or nonconflict activities such as buildup or return to peacetime conditions.

Services. Any or all of the four Services—Army, Air Force, Navy, and Marines—can be represented in the conflict. Many of the inputs have an index on Service, as is evident in the descriptions in Volume II.

---

1 Some changes to the FORCEMOB computer code would be necessary for more than four theaters to be played.

Units, or force units. A list of different types of units is specified for each Service. FORCEMOB data have generally specified the units as being the size of a battalion, air wing, or carrier battle group, but unit size is not a restriction in the model itself. For each month of conflict in each theater, inputs specify the number of units, by type, *arriving* in that theater during that month. Cumulative totals of these inputs over time represent the numbers of units *present* in a given theater during a given month. (Units are not removed from the theater, except, possibly, by attrition.)

Intensity levels. For each month of conflict in each theater, two "combat intensity levels" are input, one governing attrition and one governing consumption. These values are integers and can assume values between 1 and an input "number of possible intensity levels." Input attrition and consumption rates can vary with intensity level. In a given month of conflict, the calculations use the attrition and consumption rates that correspond to the intensity levels specified for that month.

In addition, the combat-related "items," discussed below, are essential parts of the Requirements module conflict specification.

b. Taxonomy of Combat-Related Items

The conflict-related calculations are performed using a set of combat-related "items" that can be finer grained than the Major End Items. Results are aggregated to the MEI level. We use the term "items" to encompass not just weapons themselves, but also things like ammunition, other consumables (e.g., chaff), and support functions such as operation and maintenance. FORCEMOB groups these items into the following categories.

TOE items. TOE stands for "Table of Organization and Equipment." This table delineates the types of weapons and systems that compose each of the force unit types. (FORCEMOB accounts for "TOE" items in all Services, even though not all Services use that term.) Tanks, aircraft, and ships are typical types of TOE items, but allowances for support functions (expressed in dollar terms) might also be included.

Consumption items are those items expended by a force unit during combat and which must be supplied to keep it fighting. Examples are ammunition rounds, bombs, and sonobuoys. Support functions could also be considered consumption items.

Threat items are items such as missiles and torpedoes, which are likely to be in limited supply during a conflict. Threat item usage is not explicitly tied to the number of

force units present. Rather, an extrinsic amount of threat item requirement is input. This input amount can reflect what would need to be expended against some kind of assumed or projected threat.

Each given TOE, consumption, and threat item corresponds to some Major End Item. (A given MEI might encompass several different kinds of TOE, consumption, and threat items.) These correspondences, specified in the Element database input file, are used to aggregate demand for TOE, consumption, and threat items into demand for MEIs.

c. Categories of Requirements

The requirement for combat-related items can be partitioned into the following components.

Unit startup requirements. For each unit, a list showing the number of TOE items (by TOE item type) in that unit is input. Normally, each unit entering the theater induces a set of requirements for its corresponding set of TOE items. If the user desires, however, the unit startup requirements can be set to zero via flag variables on the Control Inputs file (each theater has a different flag). This option is consistent with an assumption that all units are in existence at the start of the simulation; it can also be used for theaters that simulate non-conflict activities or later portions of a conflict. Whichever option is used, the MEI Inventories file (see section 3) should be adjusted appropriately (e.g., if startup requirements are set to zero because all necessary TOE items for units are in place, then those TOE items should not be included in the MEI inventory).

Attrition replacement requirements. As attrition occurs, additional TOE items are required to keep the units at full strength (as specified by the input lists of numbers of TOE items per unit). For each TOE item type, an input attrition rate represents the fraction of items lost in combat in a month. This rate varies with TOE item type and with the intensity level in effect for the month. From the attrition rates and the lists of TOE items, REQMOD computes losses. These losses then induce (or from a computational standpoint, become) replacement requirements. If the user desires, however, a zero-attrition replacement rule can be invoked, via flag variables on the Control Inputs file (each theater has a different flag). In this case, no attrition replacement requirements are computed, but the units do suffer attrition (and, as a result, consume less).

Consumption requirements. The number of consumption items expended in combat becomes a requirement in that that number of items must be available to be used. In each month of conflict, each unit present that month expends a set of consumption

items. Expenditures are specified by input rates (i.e., numbers of items expended per month) that vary with unit type, consumption item type, and consumption intensity level.

Threat item requirements. The number of threat items expended in combat becomes a requirement. As stated earlier, threat item usage is not tied to numbers of units present. Instead, for each threat item, a total initial requirement of that threat item is input (in terms of numbers of items), and two sets of allocation factors are input. The first allocates the total initial requirement among the different theaters. The second set consists of several subsets, one for each theater. The subset of allocation factors for a given theater allocates the requirement for that theater among the months of conflict in that theater. Together, these allocation factors allow a threat item requirement to be computed for each theater and month of conflict.

d. Determining the Total Requirement

The final results of this portion of the Requirements module are requirements, expressed in thousands of dollars, for each MEI. The Requirements module keeps track of a separate requirement amount for each combination of theater, MEI, and month of conflict. It starts by initializing these amounts to zero. Note that these results are in dollar terms, while the force structure inputs deal with numbers of combat-related items. For each combat-related item, in each category of requirements, REQMOD first computes a number of that item required (for each theater and month), as described in the preceding section. This number requirement is multiplied by an input item cost, yielding a dollar requirement.[2] This dollar requirement is then added to the total for the MEI corresponding to the combat-related item.

Note that different units of a given type (in a given theater and month) are treated identically, so that the requirement in a given category (except for threat items) is the requirement per unit multiplied by the number of units.

In accordance with the above remarks, REQMOD performs the following steps for each *played* theater:

1) From the input numbers of units arriving, compute cumulative totals over time to determine numbers of units present, by type, in each month of conflict.

---

2 For some items, such as operation and maintenance, the "number" requirement could itself be a dollar figure; in such a case, the corresponding cost would be unity.

2) For each combination of type of unit and month of conflict, the following steps are performed.

   a) Determine the unit startup requirements associated with units that arrive that month. (These are zero if the flag option has been invoked.) Convert to dollars and add to the MEI totals.

   b) Determine the attrition replacement requirements for the units present that month. (If the zero attrition replacement rule has been invoked, these requirements are zero—but in this case, REQMOD reduces the number of units.) Convert to dollars and add to the MEI totals.

   c) Determine the consumption item requirements for the units present that month. Convert to dollars and add to the MEI totals.

3) For each combination of type of threat item and month of conflict, determine the requirement for that threat item in that month. Convert to dollars and add to the MEI totals.

2. An Alternative: Direct Input of Major End Item Demand

As an alternative to computing the MEI requirements, FORCEMOB allows the MEI demand to be input.[3] We stress that this demand amount is considered to arise from the extraordinary requirement, not normal peacetime military demand. This approach was used in the Naval Wargame Logistics Support study [2] to specify reconstitution military demand.

The model requires that the MEI demand be expressed in dollars (strictly speaking, in thousands of dollars). In developing data, a user can

1. determine MEI demands as numbers of items desired.
2. do a side calculation to multiply these quantities by appropriate MEI prices to obtain dollar values to be input to FORCEMOB.

For complete consistency of results, the MEI prices can be put in the Cost database input file (see Volume II, Chapter III). All requirements and prices should be expressed in the same dollar year as the rest of the FORCEMOB monetary data.

A different MEI demand value can be input for every month within a user-specified conflict period within the scenario period. (The value represents the dollar

---

3 If the user indicates on the Control Inputs file that Optional File 4, an MEI Requirements file, is to be used, then the program assumes that the user desires to input the MEI demand rather than to have it computed from a conflict scenario. The MEI Requirements file specifies the MEI demand to be used (see Volume II).

amount of the MEI demanded that month.) There are two ways to do this. The first is to simply specify the demand amount for each MEI and month combination. The second way is to input a total MEI demand amount for each MEI, and to use a set of input allocation fractions to allocate the total demand amount over the months of the conflict period (the same set of allocation fractions is used for each MEI). For details, see the descriptions of Optional File 4 (MEI Requirements) and the Control Inputs file (run options 0b and 1b) in Volume II.

3. Major End Item Inventory

FORCEMOB allows amounts of MEI inventory to be input. These inventories can be used to offset some (or, possibly, all) of the MEI demand. Inventory is expressed in dollar terms (thousands of dollars, in the given dollar year). So, too, is the MEI demand, whether computed or input. The amelioration of demand by inventory is performed on a dollar for dollar basis.

In a multitheater conflict, a number of FORCEMOB inputs affect which theater demands can be ameliorated by the inventory. The steps are as follows:

1. For each MEI, a value for the total amount of inventory is input (via the MEI Inventories database file).
2. On the Control Inputs file, for each played theater, the user specifies a percentage value that shows the percentage of the total inventory amount that can be applied to ameliorate demand in that theater. The same percentage is used for each MEI. If desired, an optional inventory allocation file can be used to specify different percentage allocations for different MEIs.
3. From the above values, for each MEI and played theater, a pool of inventory available to be used in that theater is computed. Demands in that theater are reduced insofar as possible.
4. The user can specify (on the Control Inputs file) that certain theaters be in "inventory sharing groups." Inventory left over in a given theater (after the preceding step has been performed) can be used to satisfy demand in other theaters in the same sharing group, in accordance with an input priority ordering of theaters within the group. (See Volume II, Chapter II, for the precise way in which to specify inventory sharing groups.)

Direct MEI input case. If the MEI requirements have been input, FORCEMOB considers all demands as falling in theater 1. The inventory allocation input value (on the Control Inputs file) has the effect of letting only a certain percentage of the input

inventory pool be available. An optional inventory allocation file can be used to the same effect, on an MEI by MEI basis. Inventory sharing groups are not relevant.

Inventory and time. Although there are many rules that govern the theaters in which inventory can be applied, there are no such rules for time. Inventory is assumed to be available at the beginning of the scenario period and capable of being applied in any month of the scenario. Within a given theater, the earliest demands are ameliorated first.

4. Converting MEI Demand to Demand on Industry

Given the MEI demands net of inventory, the next step of FORCEMOB is to determine the demand on the industry sectors that is required to produce those MEIs. These calculations form the bridge between weapon demand and industry demand—and between the two modules of FORCEMOB. In some expositions of FORCEMOB (e.g., [3]), this interaction has been considered part of the Industry-level module. In terms of the current FORCEMOB computer program structure, however, it logically forms part of the Requirements module.

In a multitheater conflict, FORCEMOB allows the option that only the MEI demand arising from conflict in certain theaters is to induce industry demand. For each theater, an input flag on the Control Inputs file indicates whether the MEI demand for that theater can generate industry demand. For example, one could set up a dummy theater to draw down inventory; it might not be appropriate for such a theater to induce industry demand.

a. The Defense Translator and Total Requirements Demand

In FORCEMOB, industry demand is computed in a one-step calculation that combines two different underlying concepts. The first concept involves determining what kinds of industrial output, in what proportions, are required to make a given type of Major End Item. A data set called the Defense Translator has been developed that gives this information. For each different MEI, inputs from the Defense Translator specify the percentage of MEI demand attributable to each of the industry sectors. In other words, in order to make a certain (dollar) amount of a certain weapon system, percentage 1 of those dollars must be spent on industry 1 (i.e., constitute a demand on industry 1), percentage 2

on industry 2, and so forth. The sum of these percentages, taken over all the industries, equals 100 percent.[4]

To make their products, however, the industries specified by the Defense Translator must procure output from additional industries. Those industries, in turn, must procure output from yet more industries, and so forth. Economic input/output theory provides a methodology for estimating the "total requirements" demand on each industry arising from all such iterations.[5] This methodology allows consideration of "lower-tier" or "upstream" demands on industry, in addition to the industrial input, or "final" demand, that is directly required to build a weapon system. Some industry sectors might have no final demand yet a high total requirements demand. From publicly available data, an "input/output matrix," or "Leontief inverse matrix" can be constructed; this matrix gives the total requirements demand (in each industry sector) that is associated with one dollar's worth of final demand in any given sector (for details, see [8]).

One of the FORCEMOB input data sets is a "production process matrix" that combines the information in the Defense Translator and the input/output matrix. For each MEI and industry combination, the dollar demand for the MEI is multiplied by the appropriate element of the production process matrix to yield the demand on that industry—expressed in total requirements terms—that is induced by the demand for that MEI.

b. Production Lead Times

In calculating industry demand, FORCEMOB uses production lead times for Major End Items. If it takes m months to produce a given MEI, and there is a certain demand for the MEI at month t, then the corresponding total demand on industry is apportioned evenly over the m-month period from $t-m+1$ through t, inclusive: to produce the MEI, some industry contribution is considered necessary during every month of the lead time. Thus the production lead times have the effect of creating a time-phased demand. For example, even if all the MEI demand occurs at the end of the scenario period, the demands from industry are spread back over the production lead time, so there

---

4 For more information on the Defense Translator, see Frazier, Campbell, and Cheslow [10]. The discussions in IDA Paper P-2885 [3, Appendix B], and in the *Theoretical Foundations* volume of Paper P-2716 [8] provide additional information about the use of the Defense Translator in FORCEMOB.

5 For more information on input/output theory, see Miller and Blair [11] and the *Theoretical Foundations* volume of IDA Paper P-2716 [8].

is some industry demand earlier on in the scenario period. For each industry and month combination, the demands are summed over the MEIs to obtain a total conflict military demand on that industry during that month.

The production lead times are input to the model, via the Production Process Lead Times database file. (The FORCEMOB *Theoretical Foundations* volume of IDA Paper P-2716 [8] indicates how data for the lead times can be developed.) For ease in running sensitivity cases, FORCEMOB allows the lead times from the input file to be multiplied by a percentage factor (specified on the Control Inputs file) and the resultant values, rounded to the nearest month, used as the production lead times. (The percentage factor can exceed 100 percent, to model the effect of lengthened lead times.)

c. "Pre-Scenario" Demand

As just stated, if it takes m months to produce a given MEI, and there is a certain demand for the MEI at month t, then the corresponding total demand on industry is apportioned evenly over the m months from $t - m + 1$ through t, inclusive. However, if the lead time is long, or if the time period t is near the start of the scenario period, some of these months precede the start of the scenario period. The industry demand that would occur in these months can be called *pre-scenario* demand.

The question arises: what is the most appropriate way to model pre-scenario demand? FORCEMOB computes industry demands that are induced by the extraordinary military demand for weapons. These demands will be input to the Industry-level module. How should timing factors affect the computation of the industry demand?

By assumption, all the extraordinary military demand for MEIs occurs during the scenario period. The beginning of the scenario period marks the time when it becomes apparent that an extraordinary military demand is looming and that measures (such as expansion of industrial production) will have to be taken to meet that demand. Before the scenario period, there is presumed to be no such awareness. Thus if it takes x years to make a weapon and the scenario lasts only $x - 1$ years, then it clearly is impossible to build the weapon within the scenario period. One approach would be to report that the weapon simply cannot be produced, ignore all industry demand that it might have induced, and focus industry supply expansion toward making those items that can be produced on time. That is, the MEI demand goes unsatisfied because it cannot be produced on time. This certainly is a reasonable way of modeling MEI demand that

occurs early in the scenario, and in FORCEMOB Version 3.2, this approach is available as an option, as described in Chapter V.

The approach taken in FORCEMOB Versions 3.1 and earlier, however, is to let a proportionate part of the early MEI demand induce industry demand—the proportion equals the ratio of the demand time to the lead time. For example, if there is demand for a certain MEI two years into the scenario and the lead time for that MEI is four years, then half of that MEI demand is modeled as inducing industry demand.

A more precise description of this modeling is as follows. Suppose that the following conditions apply:

- The production lead time for a given MEI is m months.
- There is a certain demand for the MEI in month u of the scenario period, where $u < m$.
- The corresponding total requirements demand on some given industry is y.

Then

- The amount $\left(\frac{m-u}{m}\right)y$ is considered to be pre-scenario demand.
- The amount (y/m) is considered to be industry demand in month j of the scenario period, for $j = 1,\ldots,u$, inclusive.

This computation is applied to each industry that is affected by the demand for the MEI.

FORCEMOB *reports* the pre-scenario demand but does not include it in the demand values that go into the Industry-level module. The total pre-scenario demand is shown on the history file for the run. One of the output reports shows the pre-scenario demands by industry and indicates which MEI demands induce pre-scenario industry demand at which times.

The presence of pre-scenario demand indicates that the conflict military requirements profile developed by the Requirements module is, in some sense, infeasible to build within the time frame being considered. Thus in any study performed with FORCEMOB, pre-scenario demand, if present, merits careful attention, thought, and analysis.

d. The Conflict Military Requirements File

In the Requirements module, the weapon costs, MEI prices, MEI requirements, and inventories have been expressed in terms of thousands of dollars. The production

process matrix values are expressed in units of industry dollar demand per MEI dollar demand, so the resulting induced industry demand values are also expressed in terms of thousands of dollars. After the industry demands have been computed, the code divides the values by 1000, to express them in millions of dollars. These demands, for each combination of industry and month within the scenario period, become inputs to the Industry-level module (which deals in units of millions, not thousands, of dollars). As stated earlier, only the "in-scenario" demand, not the pre-scenario demand, is included.

Under some circumstances, the demands are written out to a "Conflict Military Requirements file" which encapsulates the industry demands induced by the conflict specification. This would happen in either or both of the following circumstances:

- The given run of FORCEMOB is exercising only the Requirements module, not the Industry-level module.
- The user has explicitly requested (on the Control Inputs file for the run) that a Conflict Military Requirements file be generated.

The Conflict Military Requirements file serves as a vehicle of communication between the two modules of FORCEMOB. It facilitates running the computer program. Many Industry-level module runs can be made using the same Requirements module specifications, without rerunning the Requirements module each time—the demands are simply read from the Conflict Military Requirements file.

B. INDUSTRY-LEVEL MODULE INTERACTIONS

It is reasonable to assume that during peacetime, the economy is "in balance," so that supply equals demand (civilian plus peacetime military) plus net exports. In this case, any extraordinary military demand on industry, such as that computed by the Requirements module, creates shortfalls in supply.[6] During a mobilization period prior to the conflict period (and continuing throughout the scenario period as necessary), two methods of redressing such shortfalls could be used. The first involves supply expansion through better utilization of existing industrial capacity (e.g., by adding more shifts); the second, investment in new capacity. The increased supply generated by this expanded capacity will hopefully offset the shortfalls.

---

6 The "economic balance" property is not an intrinsic requirement of the FORCEMOB model itself, but many of the data sets developed for FORCEMOB have satisfied it.

The Industry-level module (ILM) of FORCEMOB models these concepts via the following steps:

1. The conflict military demand from the Requirements module is added to civilian and base military demand to compute total demand.
2. A "pre-investment supply" value, which takes into account expanded capacity utilization (via input parameters) but not investment, is computed.
3. For each industry, the total demand is compared with supply, and shortfalls, if any, are assessed. This is done on a time period basis, so that early shortfalls cannot be satisfied by later surpluses. The "pre-investment shortfall" is reported.
4. If there are shortfalls, FORCEMOB models the process of investment to redress them. Post-investment supply and demand are then compared. Hopefully, the shortfalls have been eliminated. If not, they can be reported for further analysis.
5. The model invokes a procedure to identify times and industries where excess capacity exists, and to trim production so that no excess production occurs.

The following sections discuss these ILM interactions, in turn. Within the context of the ILM, the terms "requirements" and "demands" are used synonymously, and refer to demands on industry.

1. Computing Total Demand

a. Base Military Requirements and Factors

The base military demands are the impacts on the U.S. economy from peacetime military spending. From the standpoint of FORCEMOB, they operate in addition to the conflict military demands computed by the Requirements module. In actuality, there is not a hard and fast distinction between base military and conflict military requirements, as the base military requirements might include resources that can be used in a conflict.

Data for the base military demands are input (from the Base Military Requirements database file) for each industry and (calendar) year combination. The value for a given year is divided by 12 to obtain a monthly value, which is used for each month within the corresponding year.[7] All base military and civilian demand values are

7 The input data value represents the requirement over the whole calendar year, and one-twelfth of this value is the average requirement in a month of this year. The scenario period might start in the middle of a calendar year, but the input data value applies to the whole year, not just the portion within the

expressed in millions of (the given dollar year) dollars. For consistency with the conflict military requirements data, base military and civilian demands should be expressed in total requirements demand terms, not final demand terms. That is, the demand value for a given industry should include not only the industrial output demanded by the final demand sectors, but also the output needed by other industry sectors to make their products.

Versions 1 and 2 of FORCEMOB had the option of allowing the base military requirements to be input either by industry or by MEI. In the latter case, FORCEMOB translated the base military MEI requirements into requirements on industry via a Production Process Matrix file, as the Requirements module currently does for conflict military requirements.[8] Version 3.1, however, only allows the "by industry" format, to facilitate partitioning FORCEMOB into separate Requirements and Industry-level modules. The "by MEI" option might be reinstated in a future code version.

For ease in performing sensitivity analyses, FORCEMOB allows a number of different optional "factor" inputs. The code multiplies the initial base military requirements value by the corresponding factor value for that industry and month to produce the value that FORCEMOB will use in the demand calculations. One of these optional factors allows the base military values to be reduced during all months within the conflict period. For more information on these factors, see the description of the Base Military Requirements database file in Volume II of this paper.

b. Civilian Requirements and Factors

The civilian (or synonymously, civilian consumption) requirements encompass all nonmilitary demands on the economy, excluding imports and exports. (Data for imports and exports are input on the supply-side input data file, and are treated as described in Section 3.) Data for the civilian demands are input for each industry and year. The value for each given year is divided by 12 to obtain a monthly value. As with the base military demands, the data should be expressed in terms of total requirements, in millions of dollars.

---

scenario period. But in the ILM computations, the (computed) average monthly values are added up only over the months in the scenario period. Similar considerations apply to civilian requirements, imports, exports, and supply values.

8 For more information on this procedure, see [5] for a discussion of FORCEMOB Version 1.0 Subroutine BASECAL.

For ease in performing sensitivity analyses, FORCEMOB allows a number of different optional "factor" inputs. The initial civilian requirements value is multiplied by the corresponding factor value for that industry and month to produce the value that FORCEMOB will use. These factors can model the imposition of civilian austerity during the conflict and/or mobilization periods, by reducing the civilian demand then. For more information on these factors, see the description of the Civilian Requirements database file in Volume II of this paper.

c. Determining Total Pre-Investment Demand

For each industry and month combination, total demand is determined by adding the following components:

- The conflict military demand from the Requirements module
- The base military demand, adjusted by all relevant sensitivity factors
- The civilian demand, adjusted by all relevant sensitivity factors

This computation is performed for each month in the scenario period. FORCEMOB then computes the available supply, as described in the next section, and compares the demand with supply to determine whether any shortfalls exist.

2. Pre-Investment Supply and Modeling of Dual Use

For each industry sector and year of the scenario period, the FORCEMOB Supply-Side input data file contains pre-expansion values for domestic supply, i.e., the amount of industrial output supplied during that year under normal peacetime conditions. These yearly values are divided by 12 to obtain initial monthly values. As explained below, the model then computes expanded pre-investment supply, based on certain inputs.

The amount of supply expansion possible can be affected by the extent of civilian/military fungibility, or dual use, within an industry. An industry is considered to be dual use if its products that are used by the civilian sector and its products that are used by the military sector are similar, so that factories producing output for the civilian sector can add extra shifts or use additional capacity to produce military output. Subsections a and b explain FORCEMOB's modeling of supply expansion under conditions of complete fungibility. Then, subsection c describes how the more general case is modeled.

Methodology to model the dual use concept in FORCEMOB was developed especially for FORCEMOB Version 3. FORCEMOB Versions 1 and 2 assume complete military/civilian fungibility for all industries. Version 3.0 has the same dual use modeling capability as Version 3.1, but the input files for the relevant parameters are structured a bit differently.

a. Expanded Utilization of Plant Capacity—Complete Fungibility Case

One of the inputs to FORCEMOB is a "plant capacity utilization fraction," which refers to the product of a shift factor and a capacity utilization rate.[9] The shift factor is the fraction of a production day for which a facility is operated during peacetime. That is, if a plant is capable of operating 24 hours per day and it employs one 8-hour shift during peacetime, its shift factor is 33 percent, or 0.33. Similarly, a plant's capacity may be only partially utilized even if its work force is employed 8 hours per day—e.g., only 75 percent of available practical capacity may be occupied. Hence, during peacetime, such a facility is producing only 0.25 (= 0.75 × 0.33) of its total capacity output. Its "emergency operating capacity" (EOC) output is thus four (i.e., 1/0.25) times its peacetime output. The number 0.25 is the plant capacity utilization fraction input.

To model supply expansion, FORCEMOB uses this number as well as an additional input that represents the percentage of the gap between peacetime and EOC output which is to be closed (by adding additional shifts, lengthening the workweek, and/or better utilizing plant capacity). This "gap closure percentage" value corresponds to the percentage of spare capacity to be used. If this percentage value is, say, 10 percent and the plant capacity utilization fraction input is 0.25, then the expanded supply is

$$(1 + 0.1(4\text{–}1)\,) \times (\text{peacetime output}),$$

or 30 percent more than the peacetime output.

More generally, consider a given industry i and a given month t of the scenario period, and let θ_{it} denote the plant capacity utilization fraction input for that industry and month. Let ψ denote the proportion (i.e., the percentage divided by 100) of spare

9 FORCEMOB allows a different fraction to be input for each combination of industry sector and year, even though data might not in fact vary by year. The value for a given year is used for each month of that year. The values are input in the Q/K and Capacity Utilization Ratios database file, as described in Volume II.

capacity to be used. Then the ratio of expanded supply to peacetime output is given by the "supply expansion ratio"

$$\varphi'_{it} = 1 + \left(\frac{1}{\theta_{it}} - 1\right)\psi.$$

Note that this ratio is always greater than or equal to 1. In the absence of a ramp-up period and dual use constraints (see the following sections), the expanded supply equals this ratio multiplied by the peacetime supply.

The same value of ψ is used for all industries and months.[10] Sensitivity analyses can be performed by varying this value. Setting it to zero models peacetime conditions; setting it to 100% models the case where all industries are assumed to be operating at full EOC. The percentage value is input on the Control Inputs file.

b. Ramp-Up Period for Supply Expansion

The supply expansion can be phased in over a ramp-up period of an input number of months. Suppose that the ramp-up period is r months long. For industry i, define

$$\varphi_{it} = \begin{cases} 1 + \left(\frac{1}{\theta_{it}} - 1\right)\psi\left(\frac{t}{r}\right) & t = 1, \ldots, r-1 \\ \varphi'_{it} & t \geq r \end{cases}$$

where φ'_{it} is as defined in the previous subsection. FORCEMOB uses the quantity φ_{it}, rather than φ'_{it}, as the supply expansion ratio in month t of the scenario period. That is

$$(\text{expanded supply}) = \varphi_{it}(\text{peacetime supply}).$$

(FORCEMOB does not model the effect of possible diminishing returns when operating very close to EOC.)

The ramp-up period can model, for example, the time needed to hire all the extra labor to work the additional shifts. (Constraints on the amount of labor available are not modeled in the current version of FORCEMOB. They might be modeled in future versions.) The length of the ramp-up period, r, is input on the Control Inputs file, and the same value is (currently) used for all industries. Of course, the ramp-up period can be

---

[10] It would not be difficult to alter the FORCEMOB computer code to allow ψ to vary by industry or time period.

input as zero months or one month, in which case $\varphi_{it} = \varphi'_{it}$ for all i and t, and all the expanded supply capacity is considered to be immediately available.

c. Modeling the Civilian/Military Fungibility Level

To model various levels of dual use or civilian/military fungibility, FORCEMOB uses the following procedure, for each industry i and each month t of the scenario period:

1. Based on the plant capacity utilization fraction θ_{it}, the spare capacity utilization proportion ψ, and the ramp-up period length, determine the quantity φ_{it}, as explained in the preceding subsection.
2. Partition the pre-expansion (i.e., input, peacetime) supply value into an amount devoted to military production and an amount devoted to civilian production (see below).
3. The military portion of supply is assumed to expand to φ_{it} of its original (peacetime) value. To model fraction μ_i (i.e. $100\mu_i$ percent) civilian/military fungibility, fraction μ_i of the civilian portion of supply expands to φ_{it} of its original value; the remainder of the civilian portion of supply does not expand. The quantity μ_i is an input that can range between zero and one, inclusive. A separate value can be input for each industry i, if desired. These values are input on Optional File 6, Military/Civilian Fungibility Factors, as described in Volume II. If no such file is used, complete fungibility is assumed, i.e., $\mu_i = 1$ for all industries i.

Thus,

$$\text{expanded supply} = \varphi_{it}(\text{military supply}) + \varphi_{it}\mu_i(\text{civilian supply}) + (1-\mu_i)(\text{civilian supply}).$$

If μ_i is equal to 1 (i.e., there is complete civilian/military fungibility in industry i), then this formula is identical to the expansion formula developed previously. If μ_i is zero, then the civilian supply is assumed to be unable to expand at all, i.e., factories producing output for the civilian sector cannot add extra shifts or use additional capacity to produce military output.

FORCEMOB estimates the (pre-expansion) military supply and civilian supply as follows. The "economic balance" condition can be expressed as follows:

$$\text{pre-expansion supply} = \text{base military demand} + \text{civilian demand} + \text{exports} - \text{imports}$$

This gives rise to the formulas

military supply = base military demand + some fraction of net exports

civilian supply = civilian demand + remainder of net exports

Note that military supply plus civilian supply equals total (pre-expansion) supply. For the "fraction" of net exports, FORCEMOB uses the ratio of peacetime military demand to peacetime military plus civilian demand, i.e., the net exports are apportioned in proportion to the demand.[11] (FORCEMOB could be modified to use actual data on military and civilian supply instead of estimating these quantities, if such data become available.)

Note that even in the case of no civilian/military fungibility *at the macro level*, FORCEMOB is assuming complete fungibility between specific products within the military portion of an industry.

3. Imports, Exports, and Adjusted Supply

For each industry sector and calendar year, the FORCEMOB supply-side input data file contains values for imports and exports. To aid in sensitivity analyses, optional "import/export factors" can be applied to these values. The import factor value for a particular industry and year is multiplied by the corresponding initial import value to obtain the yearly import value that FORCEMOB will use, and similarly for exports. For more information on these factors, see the description of Optional File 3, Import/Export Factors, in Volume II. The resultant yearly values are divided by 12 to obtain monthly values.

In section B.2 above, the word "supply" referred to domestic supply. For each industry and month, the net imports value (i.e., imports minus exports) is added to the expanded domestic supply to yield a total pre-investment supply value. Conceivably, the net imports value could be negative; if so, it would reduce the supply value. But if any negative supply value results, the program prints an error message and terminates.

---

[11] The model and computer program need to be able to handle the case of general data. Thus if the data do not satisfy the economic balance condition, the military and civilian supply are computed according to the procedure just outlined, but are then adjusted so that neither value is negative and the values sum to the total peacetime supply.

4. Pre-Investment Shortfall

FORCEMOB has thus far computed total pre-investment demand and supply values for each industry in each month of the scenario period. Now, for each industry in turn, the model compares supply and demand, on a month by month basis, and computes the shortfalls in supply. Shortfalls that occur early in the scenario cannot be offset by later excesses in supply. Early excesses can be used to help offset shortfalls that occur later. The total net shortfall for each industry is reported.

For a more precise illustration, consider some given industry. Let c_t be the supply in this industry in month t of the scenario period and let d_t be the demand in month t of the scenario period. Consider month 1 first and consider the quantity

$$z_1 = c_1 - d_1.$$

If z_1 is positive, it represents a surplus in month 1, which can be carried forward to help meet future demands. If z_1 is negative, it represents a shortfall. In this case

- z_1 is added to the total shortfall amount for the given industry.
- z_1 is also added to the overall shortfall total.
- no inventory can be carried over.

We can set the amount of inventory carried over to the second month, h_1, as the positive part of z_1, i.e.,

$$h_1 = \begin{cases} z_1 & \text{if } z_1 > 0 \\ 0 & \text{if } z_1 \le 0. \end{cases}$$

For subsequent months t in the scenario period, the model computes the quantity

$$z_t = h_{t-1} + c_t - d_t$$

and sets the inventory carryover to the positive part of z_t, i.e.,

$$h_t = \begin{cases} z_t & \text{if } z_t > 0 \\ 0 & \text{if } z_t \le 0. \end{cases}$$

If z_t is negative, it represents a shortfall, which is added to the shortfall totals.[12]

---

[12] The current version of FORCEMOB, unlike Version 1.0, does not model an initial supply inventory. Thus the computations are different for the first month of the scenario period. Initial supply inventory could be reinstated in a future version of FORCEMOB. However, the concept is inconsistent with the "economic balance" assumption that supply equals demand in peacetime.

After all industries and months have been processed, the final shortfall totals are reported. (The overall total is written on the history file for the run. The industry totals are available on one of the output reports.[13]) In the fortunate event that the overall total shortfall is zero, no investment is necessary. Also, the user can request (via a flag on the Control Inputs file) that FORCEMOB not invoke its investment modeling. In either of these cases, the model skips the investment algorithm and proceeds to the supply reduction algorithm (described in section B.6, below), and then generates whatever output reports the user has requested. Otherwise, FORCEMOB models the process of investment in new capacity.

If the user invokes the "no investment" option, then the ILM-related output reports show the supply, demand, and shortfall patterns that occur if no additional investment in plant and equipment is performed. This information might be valuable to examine.

5. Investment

If shortfalls occur, FORCEMOB then attempts to compute a pattern of investment to redress them by building additional capacity. FORCEMOB assumes that all investment is in the production of military goods (to help meet the extraordinary military demand), so that lack of dual use capability within an industry does not affect the newly created supply. Newly created productive facilities are not necessarily worked at full emergency operating capacity, however. Instead, FORCEMOB assumes that they are worked at the same expanded supply level as existing facilities (fraction φ_{it} of the peacetime operating level, where φ_{it} is computed as discussed in section 2, for industry i in month t of the scenario period).

In investment modeling:

- The lead time necessary to build capacity should be taken into account.
- The industry demand generated by the process of investment (i.e., the investment demand) needs to be covered by supply.

The FORCEMOB investment algorithm attempts to find a set of times for investment and amounts of investment that will redress all remediable shortfalls, including those caused by investment demand, in a timely manner. The algorithm is discussed in detail in Chapter III. We stress that this algorithm, while being a quick and reasonable heuristic,

[13] Report 30, the supply-side summary report.

is not guaranteed to find a solution. That is, there might be a pattern of investment that can redress the shortfalls, but the algorithm might not be able to find it. In general, however, the presence of substantial shortfalls after the investment algorithm is performed indicates that the desired requirements might well be difficult or impossible to satisfy. In such a situation, the user should carefully examine and perform sensitivity analyses on the supply-side data values. It might be possible to resolve all shortfalls by such means as increased civilian austerity, dual use, or further utilization of spare capacity, in addition to investment.

6. The Supply Reduction Algorithm

a. Background

After the investment algorithm has been executed, FORCEMOB performs the supply reduction, or post-investment algorithm. The investment portion of FORCEMOB has modeled the creation of additional productive capacity. At certain times, however, there might be excess capacity. The post-investment algorithm identifies excess capacity where it occurs and determines a supply pattern, below capacity when necessary, that exactly meets demand.

Consider Figure II-1, which illustrates a supply and demand pattern (for some given industry) that might occur over the scenario period. Because of the conflict military demand, total demand builds to a peak during the middle of the conflict period. Foreseeing this peak, supply expansion and investment occur early on, and supply capacity rises to a higher level, where it remains throughout the rest of the scenario. Producing at capacity during the early part of the scenario builds up enough inventory to meet the peak demand when it occurs. However, in the latter part of the scenario, after the conflict, supply capacity exceeds demand, and there is no need to produce at capacity. For each month and industry, the FORCEMOB supply reduction algorithm computes a value for the actual supply, as opposed to supply capacity, such that

- the supply, possibly including inventory carried forward from previous months, will meet all demands on time.
- no excess production occurs.

We do want to be able to meet any secondary demand peaks in the post-conflict period, should they occur.

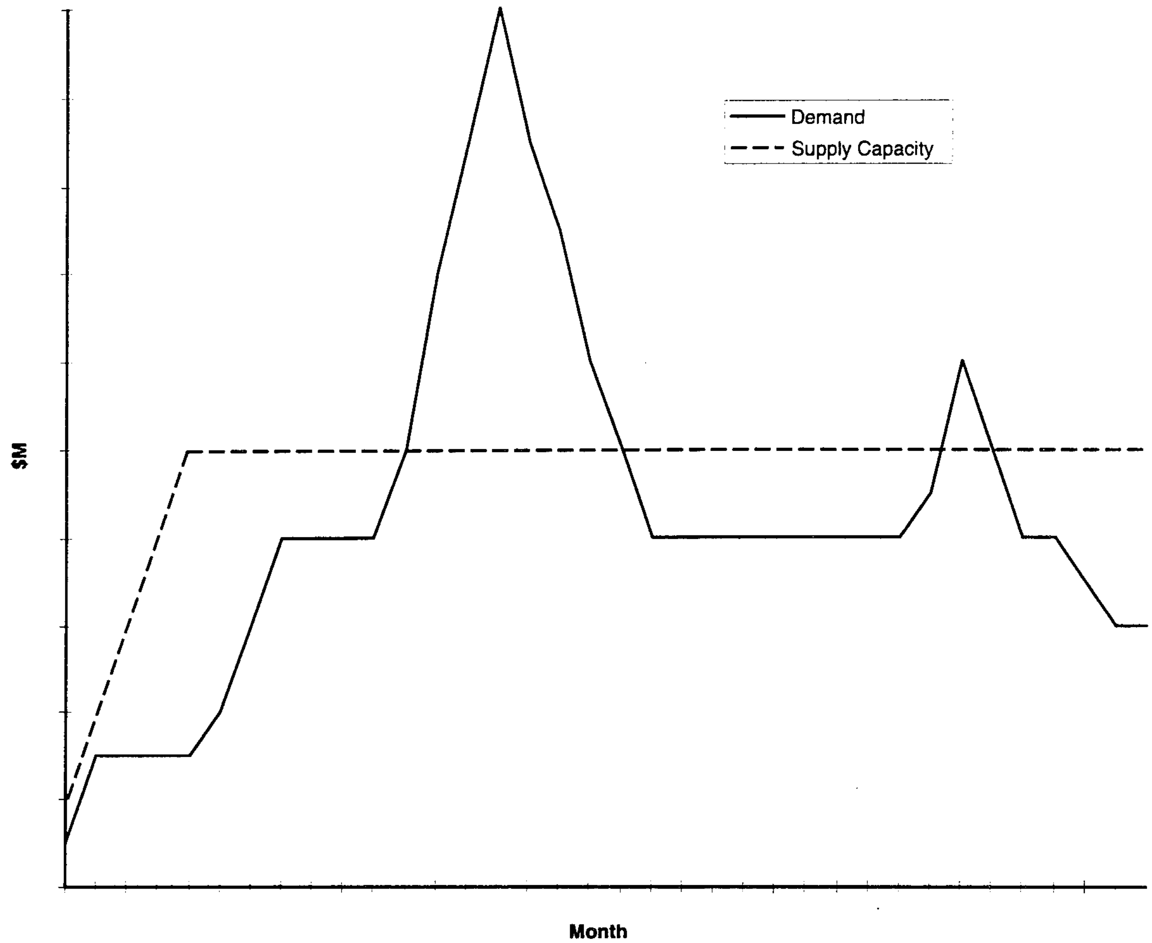

Figure II-1. Example of Supply and Demand Patterns

For certain industries, there might be irremediable shortfalls in some time periods, or the investment algorithm might not have been able to resolve all shortfalls. Even so, excess capacity might exist in later time periods, and the supply reduction algorithm identifies these and reduces supply as appropriate. Even if the user instructs FORCEMOB not to perform the investment algorithm, the supply reduction algorithm is performed.

The supply reduction algorithm gives a feel for how much of the capacity created by investment becomes excess capacity in peacetime. Another reason for the supply reduction algorithm is to aid in calculating labor requirements. Although the current version of FORCEMOB does not consider labor constraints or requirements, future versions might do so. Labor usage is keyed to actual production, not capacity.

Note that the data transferred from FORCEMOB to the Stockpile Sizing Module include military and civilian demand, investment demand, imports, and exports—but not data on industry supply. The supply reduction algorithm is irrelevant to the Stockpile Sizing Module, given the current construction and operation of the modules. The Stockpile Sizing module accepts inputs on supplies of *materials*, rather than industrial output.

b. Summary of the Algorithm

The supply reduction algorithm is performed separately for each industry in turn. The algorithm starts by transferring to working arrays the supply capacity and demand values for each month of the scenario period. If the investment algorithm has been performed, these are the post-investment values. Supply capacity values are never negative, but some demand values might be (one interpretation is that the civilian populace willingly gives up goods to the war effort). Negative demand values are treated as increments to the supply capacity.

The main part of the supply reduction algorithm has three parts, or passes:

1. The first pass identifies a month, t_1, such that in months 1 through t_1 of the scenario, it is necessary to produce at full capacity in order to meet all demand in that period.
2. The second pass identifies a month, t_2, and a fraction α. In months t_1+1 through t_2 of the scenario, setting the supply to the fraction α (i.e., 100α%) of the capacity will exactly meet demand in those months. Producing at a fixed percentage of capacity avoids great variances in production from month to month.
3. The third pass determines a supply pattern in the months after t_2 of the scenario period that will meet all demand in these months.

Depending on the particular supply and demand patterns, t_1 might occur at the end of the scenario (so passes 2 and 3 are not performed) or t_2 might equal t_1 (i.e., there is no time span where production is cut as indicated in pass 2).

Note that the supply capacity value that is input to the supply reduction algorithm *includes net imports*. These are subtracted from the supply value computed by the three passes to yield a final domestic production value. (Adjustment for negative demands, if any, is also made at this stage.)

c. Output

The output reports on supply expansion show the supply at all points of the simulation:

- Initial peacetime, before supply expansion
- After supply expansion but before investment
- After investment but before the supply reduction algorithm
- After the supply reduction algorithm

The output reports on supply and demand show the final supply value, after the supply reduction algorithm has been performed.

The Debugging Flags input file (see Volume II, Chapter III) has an option that will generate somewhat detailed printout about the supply reduction algorithm results.

III. THE FORCEMOB INVESTMENT ALGORITHM

Given that there are shortfalls in supply, FORCEMOB models the process of remediating these shortfalls by investing in new plant capacity. The investment algorithm is a major part of the FORCEMOB model. The investment modeling in the current version of the FORCEMOB model differs considerably from that of Version 1.0. As documented in [8], the equations of the supply-side modeling *were* based on the Cobb-Douglass production function; now a simpler linear model is used. This change makes the model easier to use and its results easier to understand.

This chapter first reviews some issues relevant to investment modeling. It then presents a nontechnical description of the investment algorithm. Finally, it presents a detailed, precise description of the algorithm, using mathematical notation.

A. ISSUES AND FORMULATION

1. Investment Lead Times and Irremediable Shortfall

The process of building additional capacity in an industry (e.g., building a factory to produce aircraft) takes place over a period of time called the investment lead time. Do not confuse the investment lead times with the production process lead times for MEIs (Chapter II, section A.4.b). The investment lead times are inputs to the model. They can vary by industry.

No investment can be commenced before the start of the scenario period (because there is assumed to be no knowledge of the necessity to prepare for an extraordinary demand on the economy). Thus if the investment lead time in industry i is l_i, any investment in industry i will be completed after month l_i of the scenario period, i.e., during or after month (l_i+1) of the scenario period. FORCEMOB can also model a general delay on investment. An input, call it b, specifies the earliest month of the scenario period that an investment—in *any* industry—can be completed and ready to start producing additional output. (Of course, b can be set to zero if desired.) Then any

shortfall in industry i that occurs before month $\max\{b, l_i+1\}$ cannot be redressed by investment—it is *irremediable.*[1]

The pre-investment irremediable shortfall is reported.[2] Its presence indicates that shortfalls cannot be completely addressed by investment given the current data and that there needs to be careful attention to and sensitivity analysis upon the values of investment lead times, capacity utilization and supply expansion fractions, dual use, supply, and demand. Note that the total pre-investment shortfall is not dependent on the values of the investment lead times—only on the supply and demand. The pre-investment *irremediable* shortfall, however, *is* dependent on the values of the investment lead times. Even if pre-investment irremediable shortfall occurs, the FORCEMOB investment algorithm is performed to attempt to satisfy the remediable shortfalls.

2. Investment Demand

The process of building new capacity in a given industry, call it industry i, involves procuring output from a number of different "feeder" industries over a period of time (the investment lead time) and using this output to build factories, machinery, and the like that can produce the products of industry i. For instance, to build a factory to produce aircraft, one needs, among other things, concrete and machinery. For each combination (i,j) of industries, the FORCEMOB input databases contain a "capital coefficient" that gives the amount of such investment demand on industry j induced by a dollar of investment in industry i. Investment demand constitutes an additional source of demand on industry, and the total demand on an industry includes the military and civilian demand, computed earlier in the model, plus any demand induced by investment.

As with the other sources of demand in FORCEMOB, the investment demand is expressed in terms of total requirements (see the discussion in Chapter II, section A.4). "Investing" x dollars in industry i means purchasing a total of x dollars worth of goods from the feeder industries that immediately contribute to building capacity in industry i. The x dollars are apportioned among these feeder industries. However, to make their products, these industries have to purchase outputs from other industries, which in turn

---

1 Let us call $u_i = \max\{b, l_i+1\}$ the "minimum investment completion month" for industry i. We will refer to the minimum investment completion month later on in this discussion.

2 In this context, the phrase "pre-investment" simply means "before the FORCEMOB investment algorithm is performed." Similarly, "post-investment" quantities are those that occur after the investment algorithm has been performed.

need to purchase outputs from other industries, and so forth. The input capital coefficients give the *total requirements*-induced investment demand, encompassing all subtiers.

This investment demand occurs over the investment lead time—and has the potential to create additional shortfalls. Such shortfalls may be irremediable if they occur early enough in the scenario period. A full resolution of shortfalls must include resolution of the shortfalls caused by investment demand, as discussed in the next subsection.

3. Supply/Demand Tradeoffs: The FORCEMOB Investment Problem

Every investment has an associated time frame. Investing x dollars in industry i at a given time means spending a total of x dollars on the appropriate feeder industries, over the lead time period consisting of months $t-l_i$, $t-l_i+1,\ldots,t-1$, so that the investment is completed and x dollars worth of additional capacity is in place at the *beginning* of month t. (The quantity l_i is the investment lead time for industry i.) As noted earlier, no investment in industry i can be completed before the minimum investment completion month u_i, defined as the maximum of l_i+1 and an extrinsic input b.

An investment in industry i that is completed at the beginning of month t increases the amount that industry i can produce during and after month t. However, it also creates an investment demand on the feeder industries, and this demand occurs before month t, i.e., earlier in the scenario. This increased demand has the potential to create shortfalls in the feeder industries—and if those shortfalls occur early enough, they will be irremediable. Every investment involves such a supply/demand tradeoff, and the problem for FORCEMOB is to determine a pattern of investment amounts and timings that will rectify all (remediable) pre-investment shortfalls without creating any additional ones.[3]

Ultimately, the problem of determining this pattern of investment can be formulated as a mathematical programming problem.[4] The appendix presents such a

---

3 Of course, if some of the pre-investment shortfalls occur so early as to be irremediable, then the investment algorithm cannot redress them. The pre-investment irremediable shortfalls are reported, but then ignored: the investment algorithm attempts to redress the remediable pre-investment shortfalls.

4 Given the linear relation of output to investment that the current version of FORCEMOB assumes, the investment problem can be formulated as a linear programming problem. Nonlinear formulations might be able to model a wider variety of output/investment functions.

formulation. Mathematical programming algorithms can then either find a pattern if one exists or prove that no such pattern exists.

FORCEMOB does not attempt to solve a mathematical programming problem. (To embed mathematical programming algorithms in the FORCEMOB computer program would be prohibitively time-consuming and cumbersome.) Instead, FORCEMOB uses a heuristic algorithm to determine a pattern of investment amounts and timings. Essentially, it looks for the remediable shortfalls that occur fairly early in the scenario and makes investments such that the increased supply capacity will redress those shortfalls (and help offset later shortfalls, if any). It is hoped that these investments don't induce any irremediable shortfalls in the feeder industries, but if they do, the induced shortfalls are reported but are otherwise ignored. Thus there can be a "post-investment irremediable shortfall," which encompasses the shortfalls induced by the investments.[5]

The fact that the FORCEMOB algorithm generates post-investment shortfalls only in certain cases makes one wonder whether there might be an investment pattern that does not generate such shortfalls. Mathematical programming methodology can resolve this issue. However, the presence of post-investment shortfalls should certainly signal the analyst to carefully examine and perform sensitivity analyses upon all the data on demand, supply, and investment.

With a clear understanding of the foregoing modeling issues, we can now turn to the algorithm itself. Section B explains the steps of the FORCEMOB investment algorithm in nonmathematical terms. Then, section C presents a mathematical description of the algorithm. When reading these descriptions, keep in mind the following points:

- All monetary quantities concerning supply, demand, shortfall, and investment are expressed in millions of (the given dollar year) dollars.
- In the context of FORCEMOB, the only distinction between a "remediable" shortfall and an "irremediable" one is the time period in which it occurs: a shortfall in a given industry is irremediable only if it occurs before the minimum investment completion month for that industry.
- In describing the investment algorithm, the term "supply" refers to the supply *capacity*, i.e., the maximum amount an industry can supply (when operating

5 Plus the pre-investment irremediable shortfall, if any. Note that the post-investment irremediable shortfall will always be greater than or equal to the pre-investment irremediable shortfall.

at the specified percentage of EOC). In some time periods of the scenario, it might not be necessary to produce at full capacity. In the supply reduction section of FORCEMOB, discussed in Chapter II, section B.6, FORCEMOB does distinguish between supply capacity and the actual amount supplied.

- FORCEMOB considers time as running month by month, through the scenario period. Conceivably, a different time period unit could be used, subject to availability of appropriate data. Changing the time period unit would involve some recoding of the FORCEMOB computer program.
- The results may show some post-investment *remediable* shortfall, but it is less than a certain tolerance fraction (currently 0.1%) of the demand. Smaller tolerance fractions have led to problems in the operation of the computer program due to limits on numerical accuracy. The phrase "significant shortfall" indicates a shortfall that is greater than this tolerance.

B. NONMATHEMATICAL DESCRIPTION OF THE ALGORITHM

1. Introduction

This section presents a nonmathematical description of the investment algorithm. The algorithm consists of a sequence of iterations. It stops either when all remediable shortfalls are less than a certain percentage of demand (currently set to 0.1%) or when an input maximum number of iterations has been executed. Each iteration of the algorithm has two main parts:

Part 1. Construct a list of industries with remediable shortfalls and, for each industry on the list, identify the earliest time period a remediable shortfall occurs (step 2 in section B.2).

Part 2. For each industry on the list, determine an investment pattern such that the supply increase from this investment will exactly meet the shortfall in the time period identified in part 1. Compute the increased supply and the investment demand induced by this investment (steps 3 through 7 in section B.2).

The supply increases and investment demands are added to the base supply and demand. These new supply and demand values then become inputs to the next iteration of the algorithm.

A flowchart of the algorithm appears in Figure III-1. The reader might wish to refer to this flowchart while reading the algorithm description here and the more mathematical description in section 3.

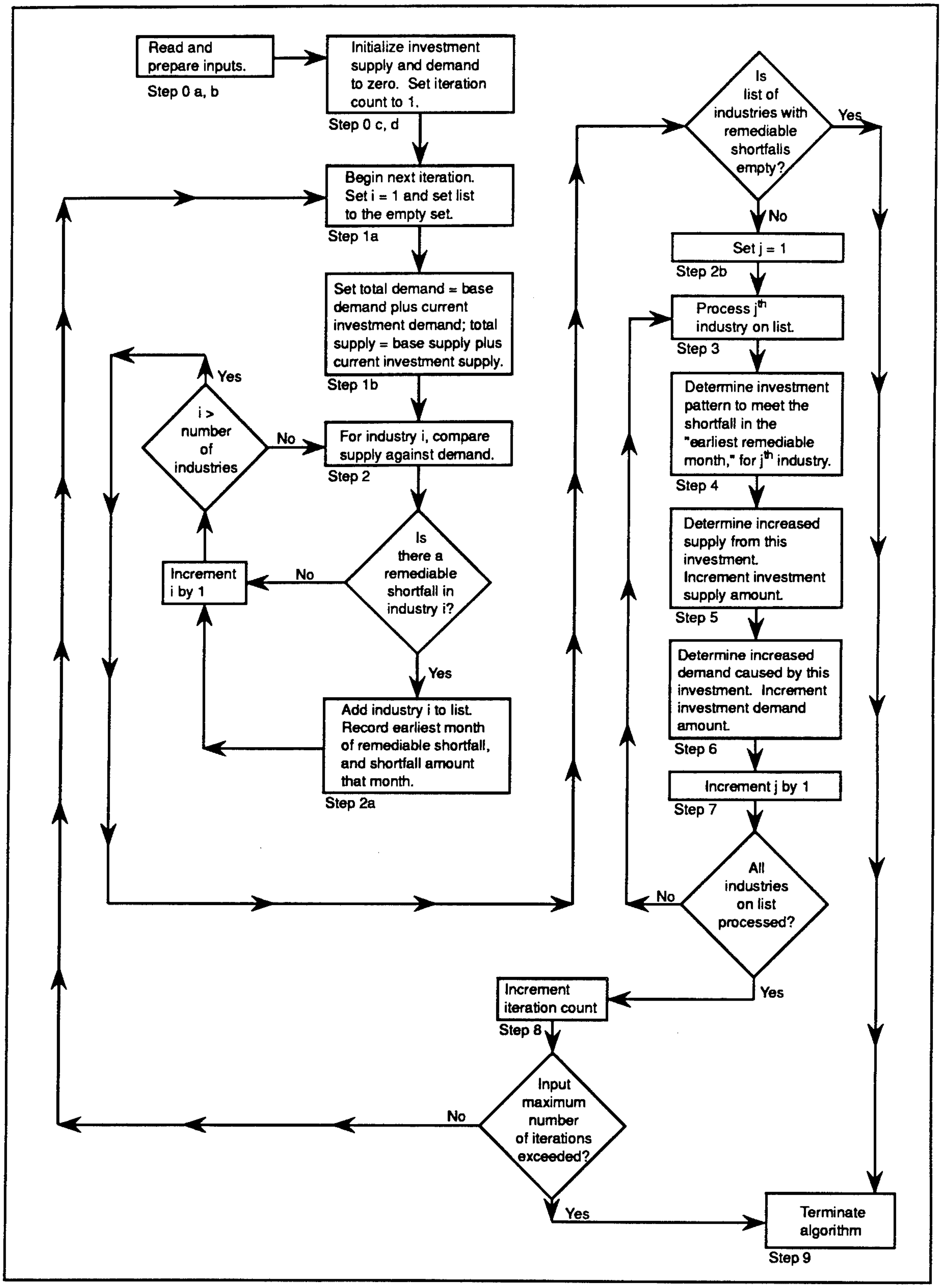

Figure III-1. Flowchart of FORCEMOB Investment Algorithm

2. Steps of the Algorithm

Step 0. Initialization

a) Read in data on capital coefficients and investment lead times. Compute adjusted lead times by multiplying the inputs from the data file by a factor specified on the Control Inputs file.

b) For each industry, compute its minimum investment completion month, based on the adjusted investment lead time. An investment in a given industry cannot be completed before the beginning of that industry's minimum investment completion month.

c) Initialize the demand to the sum of the conflict military, base military, and civilian demands. Initialize the supply to the adjusted supply value that was computed as described in Chapter II. This value includes net imports and considers partial or total expansion to EOC. Initialize the demand increments and supply increments to zero, for each industry and month.

d) Set the iteration counter to 1.

Step 1. Compute total supply and demand for current iteration

a) Set the "investment list" for the current iteration to the null set.

b) For each industry and month, set the total demand as the sum of the initial demand and the current value of the demand increment. The demand increments constitute the investment demand generated so far. For each industry and month, set the total supply to the initial supply plus the current value of the supply increment. The supply increments reflect the output resulting from the additional capital that has been generated via investment (so far).

c) Note that steps 1 through 7 apply to the current iteration of the algorithm.

Step 2. Determine the "investment list" of industries with remediable shortfalls

a) For each industry in turn, compare total supply against total demand, on a month by month basis. If supply is greater than demand, the surplus can be used to help offset demands at later times. If demand is greater than supply plus previously accumulated surplus, a shortfall occurs. We look for the earliest month *during or after*

the minimum investment completion month that has a significant shortfall. Make note of the month and the amount of shortfall, and add the industry to the investment list for this iteration. (Shortfalls that occur before the minimum investment completion month are considered irremediable. They are noted, totaled up, and reported, but the model does not attempt to redress them.)

b) If there are no industries on the list, i.e., there is no industry with a remediable significant shortfall, the algorithm is completed; go to step 9. Otherwise, set the "current industry" to the first industry on the list and go to step 3.

Step 3. Model investment in the current industry on the investment list

Consider the current industry on the investment list. We model the process of investment to remedy the shortfall in this industry in the month noted in step 2. Steps 4, 5, and 6, below, apply to this industry.

Step 4. Determine amount of investment

To remedy the shortfall, FORCEMOB computes an investment pattern that satisfies the following conditions:

- A sequence of investments is made.
- Each investment involves the same amount of money.
- The first investment is completed at the beginning of the minimum investment completion month (the investment takes place over the lead time preceding this month). The second investment is completed at the beginning of the month after the minimum investment completion month. The third investment is completed at the beginning of the month after that, and so forth. The last investment in the sequence is completed at the beginning of the month in which the shortfall occurs.
- Each investment starts producing output immediately after it is completed.
- The output from these investments that has accumulated by the end of the shortfall month is just sufficient to cover the shortfall.

These assumptions serve to mitigate great variations in the amount of investment. (In general, the timings of the investments can have significant effect on the overall supply and demand patterns, and careful attention to the timing issue is warranted.)

Step 5. Determine increase in supply

An investment completed at the beginning of a given month is assumed to start producing output (i.e., supply) in that month and every month thereafter. Under peacetime operating conditions, the monthly output from an investment is given by the amount of the investment multiplied by the *monthly* capital/output ratio for the particular industry under consideration (these ratios are inputs to FORCEMOB, and correspond to one-twelfth of the standard capital/output ratios, which are measured in yearly terms).[6] If partial or total expansion to EOC has taken place, the output produced is assumed to equal the EOC expansion fraction times the peacetime output.

The supply increase in a given month is a function of all the investments that have been completed at the beginning of that month or earlier. The supply increases arising from this iteration of the algorithm are added to the total supply increment for the industry under consideration.

Step 6. Determine increase in investment demand

An investment of x dollars in industry i is assumed to induce an investment demand on industry j of $K_{ji}x$ dollars, for each industry j. The capital coefficients K_{ji} are inputs to FORCEMOB. If the investment is completed at the beginning of month t, then FORCEMOB assumes that the associated investment demand is spread evenly over the l_i months prior to t, where l_i is the investment lead time for industry i. (This assumption is reasonable, given the absence of precise information on the flow of investment demands that occur in the process of expanding the capacity of an industry.) Step 4 has determined a sequence of investments. The investment demands induced by these investments are added to the demand increments for the current iteration. All the industries j for which K_{ji} is nonzero will be affected.

Step 7. Treat next industry for the current iteration

Consider the next industry on the investment list for the current iteration. If all industries on the list have been treated, go to step 8. Otherwise, go to step 3.

---

6 The term "output/capital ratio" seems more sensible in this context than "capital/output ratio," as we are talking about an amount of output produced per unit of capital in place. However, the term "capital/output ratio" is commonly used.

Step 8. Go on to the next iteration of the algorithm

Increment the iteration counter and go to step 1. (If more than an input maximum number of iterations have been executed, a special exit from the algorithm occurs. To prevent this from happening, set this input to a high value.[7]) Note that steps 4 through 6 have been performed for *each* industry on the investment list for the current iteration before the iteration ends and the total supply and demand are recomputed. Therefore, the order in which the industries are treated on a given iteration does not affect the supply, demand, and shortfall patterns on subsequent iterations.

Step 9. Exit the algorithm

No more significant remediable shortfalls exist. Record the resultant demand, supply, amount of investment, total shortfall (if any), and irremediable shortfall (if any). Exit the algorithm.

C. MATHEMATICAL DESCRIPTION OF THE ALGORITHM

To describe the investment algorithm precisely, mathematical notation is helpful. This section recapitulates the algorithm description of section B.2 in mathematical terms.

1. Notation

a. Indices

i, j = indices for industry

t, τ = indices for month (time period)

m = index for iteration of algorithm

r = index for industry on list of industries with remediable shortfalls, for a given iteration

b. Constants

$d_{it}^{(0)}$ = base demand (before the investment algorithm is performed) on industry i in period t. Encompasses civilian, base military, and conflict military demand.

$c_{it}^{(0)}$ = base supply (before the investment algorithm is performed) in industry i in period t. Encompasses domestic supply, adjusted by the capacity utilization

7 The user might wish to have the algorithm stop after a few iterations, to examine the interim results.

expansion factor φ_{it}, plus net imports. Allows for dual use as specified in the input data.

K_{ji} = capital coefficient that gives the number of dollars of investment demand on industry j that is induced by one dollar worth of investment in industry i, i.e., element (j,i) of the investment distribution matrix. Note that for FORCEMOB, this value should be in total requirements terms, i.e., it should encompass not only the goods purchased by the investment but also the "lower tiers" of goods purchased, as discussed in section A.2. In the FORCEMOB computer program, values $K'_{j\pi}$ are input for each of a number of different investment patterns π (via the Investment Distribution input file) and the investment pattern associated with industry i is input via the Investment Sector Mapping file. The value $K'_{j\pi_i}$ is then used as K_{ji}.

l_i = the investment lead time associated with industry i. The FORCEMOB computer program accepts values l'_i from the Investment Lead Times file and a percentage multiplier 100λ from the Control Inputs file. The code then sets $l_i = \max\{\lfloor \lambda l'_i + .5 \rfloor, 1\}$.

b = an extrinsically-specified lower limit on the time that investment can be completed in any industry—i.e., no investment in any industry can start producing additional output before month b of the scenario period. (The input value for b can be zero.)

u_i = minimum investment completion month for industry i, defined as $u_i = \max\{b, l_i + 1\}$. No investment in industry i can be completed, ready to produce additional output, before the beginning of month u_i of the scenario period.

ρ_i = the amount of output that a dollar's worth of capital produces during a month for industry i under peactime conditions. This value is input from the databases. It can be thought of as being equal to one twelfth of the standard capital/output ratio for the industry, which is measured in yearly terms.

ε = a tolerance factor for shortfall. The model will not attempt to remedy shortfalls that are less than the proportion ε of the demand (for any given industry and month). This is done to avoid problems with numerical roundoff errors in the operation of the computer program. Currently, ε is hard-coded in the computer program at .001 (0.1%).

φ_{it} = the "EOC supply expansion factor" for industry i in period t, computed as described in Chapter II, section B.2.

T = number of months in the scenario period.

M = an upper limit on the number of iterations of the algorithm to be performed.

c. Global Working Variables

$\Delta_{it}^{(m)}$ = total investment demand on industry i in period t that has accumulated by the end of iteration m

$d_{it}^{(m)}$ = total demand (base plus investment demand) on industry i in period t at the beginning of iteration m

$\Gamma_{it}^{(m)}$ = total additional supply in industry i in period t, generated by new investment, that has accumulated by the end of iteration m

$c_{it}^{(m)}$ = total supply (base plus additional) on industry i in period t at the beginning of iteration m

d. Iteration-Specific Working Variables

These working variables apply to a given iteration of the investment algorithm, and are redefined for each iteration. Additional iteration-specific working variables are defined in the course of the algorithm description.

h_{it} = inventory or shortfall in industry i during time period t

R = number of industries that have significant remediable shortfalls

ι_r = (index of) the r^{th} industry on the list of industries with (significant) remediable shortfalls (i.e., industry ι_r is the r^{th} industry on the list)

v_r' = month of earliest (significant) remediable shortfall for the r^{th} industry on the list

w_r' = amount of such shortfall

v_i = shortfall period, indexed by industry; defined by $v_i = v_r'$ if $i = \iota_r$, and not defined for other i

w_i = shortfall amount, indexed by industry; defined by $w_i = w_r'$ if $i = \iota_r$, and not defined for other i

2. Steps of the Algorithm

Step 0. Initialization

Be sure that all the constants have been computed and are at hand.

Initialize m, the iteration counter, to 1.

Initialize the investment demand, $\Delta_{it}^{(0)}$, and the investment supply, $\Gamma_{it}^{(0)}$, to zero, for all industries i and months t.

Step 1. Compute total supply and demand for current iteration, iteration *m*

At the outset of the iteration, keep the investment demand and supply at their most recent values, i.e., set

$$\Delta_{it}^{(m)} = \Delta_{it}^{(m-1)} \text{ and}$$

$$\Gamma_{it}^{(m)} = \Gamma_{it}^{(m-1)}.$$

Update the total demand and supply to include the investment supply and demand generated so far, i.e., define

$$d_{it}^{(m)} = d_{it}^{(0)} + \Delta_{it}^{(m-1)} \text{ and}$$

$$c_{it}^{(m)} = c_{it}^{(0)} + \Gamma_{it}^{(m-1)}.$$

(Recall that Γ and Δ are *cumulative* totals.)

Set R, the number of industries with remediable shortfalls, to zero; also set the counter r to zero.

Set the initial supply inventory h_{i0} to zero (FORCEMOB could be changed to allow some input amount of initial supply inventory).

Step 2. Determine the "investment list" of industries with remediable shortfalls

For ease in notation, we suppress the superscript (m) and simply consider the current values of supply c_{it} and demand d_{it}. For each industry i, in turn, the following "substeps" are performed:

a) For each month t from 1 through $u_i - 1$, inclusive, compute the inventory or shortfall value

$$h_{it} = h_{i,t-1}^{+} + c_{it} - d_{it},$$

where the "+" superscript denotes the positive part function, i.e., for a general x,

$$x^{+} = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

If the quantity h_{it} is positive, then there is a surplus or inventory in period t. FORCEMOB assumes that this inventory can be used to help offset future demands. A negative value of h_{it} corresponds to a shortfall in period t. Here, t is less than u_i, so this

shortfall is irremediable; it is collected and reported, but no further action occurs. The quantity h_{i,u_i-1} is needed for substep b), as follows.

b) For each month t, starting at u_i and proceeding forward, compute the surplus or shortfall

$$h_{it} = h^{+}_{i,t-1} + c_{it} - d_{it},$$

as before. If there is a surplus, continue to the next month t. If there is a shortfall, it is remediable, since $t \geq u_i$. First, the algorithm checks if the shortfall is within tolerance. That is, if

$$|h_{it}| < \varepsilon d_{it},$$

then the shortfall is collected and reported, but computation then proceeds to the next time period. Otherwise, i.e., if there is a significant shortfall in period t, then industry i is added to the investment list for this iteration via the following actions:

R is incremented by 1

r is incremented by 1

ι_r is set to i

v'_r is set to t

w'_r is set to $|h_{it}|$

v_i is set to t

w_i is set to $|h_{it}|$

Note that the value of v_i is generally greater than its value on previous iterations because the algorithm would have ameliorated earlier shortfalls on earlier iterations. The value of v_i could decrease from the previous iteration if the previous iteration generated a large investment demand on industry i.

Substeps a) and b) are then performed for the next industry (values h_{it} are not computed for t beyond v_i). If, for a given industry, the end of the scenario period is reached without encountering any significant shortfalls, substeps a) and b) are repeated for the next industry, and so forth.

After all the industries have been processed, if the value of R has remained zero, i.e., there are no significant shortfalls, then the algorithm terminates. The current values

of $\Delta_{it}^{(m)}, \Gamma_{it}^{(m)}, d_{it}^{(m)}$, and $c_{it}^{(m)}$ become their final values and are used in the remainder of the FORCEMOB run. Otherwise, the code sets $r = 1$ and proceeds to step 3.

Step 3. Model investment in the current industry on the investment list

Consider the r^{th} industry on the investment list—this is industry ι_r, which for simplicity we denote i. We model the process of investment to remedy the shortfall w_i in this industry in month v_i. Steps 4, 5, and 6, below, apply to this industry.

Step 4. Determine amount of investment

To remedy the shortfall, FORCEMOB computes an investment pattern that satisfies the following conditions, as stated earlier:

- A sequence of investments, each of x dollars, is made.
- The first investment is completed at the beginning of month u_i (the minimum investment completion month). The second investment is completed at the beginning of month $u_i + 1$, and so forth. There are $v_i - u_i + 1$ investments, and the last one is completed at the beginning of month v_i, the month of the shortfall.
- Each investment starts producing output immediately after it is completed.
- The output from these investments that has accumulated by the end of month v_i is just sufficient to cover the shortfall.

Under these circumstances, the amount x is given by $S / \rho_i \varphi_{iv_i}$, where the intermediate variable S is defined by $S = 2w_i / [(v_i - u_i + 1)(v_i - u_i + 2)]$. The rationale behind this formula for x appears in section D.1, below. The formula depends on the amount of shortfall, the time of the shortfall, the capital/output ratio, and the EOC supply expansion factors.

Step 5. Determine increase in supply capacity

An investment completed at the beginning of a given month is assumed to start producing output (i.e., supply) in that month and every month thereafter. Under peacetime operating conditions, the monthly output from an investment x in industry i is given by $\rho_i x$. If partial or total expansion to EOC has taken place, the output produced equals the EOC expansion factor times the peacetime output.

The investment pattern involves an investment x being completed at (the beginning of) each month between u_i and v_i, inclusive. The total additional investment in place at the beginning of month t arising from this pattern is then

$$y_t = \begin{cases} 0 & t < u_i \\ (t - u_i + 1)x & u_i \le t \le v_i \\ (v_i - u_i + 1)x & t = v_i + 1, \ldots, T. \end{cases}$$

The model uses the formula Sy_t / x for the output in month t generated by the investment pattern. The investment supply is updated for each month:

$$\Gamma_{it}^{(m)} \leftarrow \Gamma_{it}^{(m)} + (Sy_t / x), \quad t = u_i, \ldots, T,$$

where the symbol "←" can be read "is replaced by" and indicates incrementation.

Step 6. Determine increase in investment demand

An investment of x dollars in industry i is assumed to induce an investment demand on industry j of $K_{ji}x$ dollars, for each industry j. FORCEMOB assumes that this investment demand is spread evenly over the investment lead time l_i, so that if the investment x is completed at the beginning of month t, an investment demand of $K_{ji}x/l_i$ occurs in each of the l_i months prior to t. (This assumption is reasonable, given the absence of precise information on the flow of investment demands that occur in the process of expanding the capacity of an industry.) Let $\eta'_{j\tau it}$ denote the investment demand on industry j in month τ arising from such an investment; then $\eta'_{j\tau it}$ is given by

$$\eta'_{j\tau it} = \begin{cases} K_{ji} x / l_i & \tau = t - l_i, \ldots, t-1 \\ 0 & \text{for all other } \tau. \end{cases}$$

The investment pattern under consideration involves completing an investment of amount x at the beginning of each month t between u_i and v_i, inclusive. The investment demand on industry j during period τ that is induced by this pattern is then

$$\eta_{j\tau i} = \sum_{t=u_i}^{v_i} \eta'_{j\tau it} \ .$$

A closed-form expression for $\eta_{j\tau i}$ is derived in section D.2, below. The investment demand $\Delta_{j\tau}^{(m)}$ is incremented by $\eta_{j\tau i}$, for each j and τ where this increment is nonzero. (This procedure is considerably faster than performing a separate incrementation of $\Delta_{j\tau}^{(m)}$ for each $\eta'_{j\tau it}$.)

Step 7. Treat next industry for the current iteration

Consider the next industry on the investment list for the current iteration, i.e., increment the counter r by 1. If all industries on the list have been treated, i.e., if $r > R$, then go to step 8 (next iteration). Otherwise, go to step 3 to model investment in industry ι_r for this iteration.

Step 8. Go on to the next iteration of the algorithm

Increment the iteration counter m by unity. If m is now greater than M, a special exit from the algorithm occurs (set M high to prevent this from occurring). Otherwise, go to step 1 for the next iteration of the algorithm.

Step 9. Exit the algorithm

When this step is reached, any remediable shortfalls that exist are "insignificant," i.e., they are within the tolerance fraction ε of the demand. FORCEMOB records some information for further use, including:

- Final values of total demand and supply, for each industry and month
- Final values of the investment demand, for each industry and month
- Final values of the amount of investment, for each industry and month
- Final values of the irremediable shortfall, if any, by industry.

After recording this information, FORCEMOB exits the algorithm.

D. MATHEMATICAL DETAILS

This section can be omitted without loss of continuity.

1. Computation of the Investment Amount

This section provides further detail on step 4 of the investment algorithm. Given the assumptions of the algorithm, it derives the amount of investment necessary to meet a certain shortfall in a certain industry on a given iteration of the investment algorithm. All notation is the same as in previous sections. Assume that we are trying to meet a shortfall of w_i in industry i in period v_i.

As stated earlier, the investment algorithm considers a sequence of staggered investments, each of size x, with completion times ranging from (the beginnings of) periods p through v_i inclusive. (In the algorithm as currently implemented, p is set to the

value u_i, but the formulas derived here work for more general p, as long as $u_i \le p \le v_i$. It might be interesting to explore the effects of varying p.) An investment completed at the beginning of period t implies that a corresponding amount of additional capacity is fully available at the start of period t, ready to produce additional output in periods t and beyond. We wish to determine the value of x such that the total accumulated output from the investments, over the time span p through (the end of) v_i, will equal the demand w_i.

Suppose investment is made as described. Then the new investment in place (i.e., complete) at the beginning of period p is equal to x, the new investment in place at the beginning of period p+1 equals $2x$, the new investment in place at the beginning of period p+2 equals $3x$, and so forth. More formally, if y_t denotes the new investment in place at the beginning of period t, then

$$y_t = \begin{cases} 0 & t < p \\ (t-p+1)x & t = p, p+1, \dots, v_i \\ (v_i - p + 1)x & t = v_i + 1, v_i + 2, \dots, T. \end{cases}$$

If the new investment in place at the beginning of period t is equal to y_t, then the output in period t caused by the additional investment is equal to $\varphi_{it}\rho_i y_t$. The interpretation is that in peacetime conditions, the output would simply equal $\rho_i y_t$, the monthly capital/output ratio times the new investment in place. At higher operating levels, as specified by the EOC supply expansion factor φ_{it} for period t, output is φ_{it} times the base output.

Thus the given investment pattern causes $\varphi_{it}\rho_i(t-p+1)x$ additional amount of output, for each time t from p through v_i, inclusive. We want the total output over this time span to equal w_i, the shortfall, i.e., we desire

$$w_i = \sum_{t=p}^{v_i} (t-p+1)\varphi_{it}\rho_i x.$$

Thus the desired investment per period, x, is given as

$$x = w_i \Big/ \left(\rho_i \sum_{t=p}^{v_i} (t-p+1)\varphi_{it} \right).$$

If all φ_{it} have the same value, φ_i, for all relevant t, then the indicated sum becomes equal to $\varphi_i(v_i - p + 1)(v_i - p + 2)/2$, yielding a somewhat simpler formula for x.

For coding simplicity, the model makes this approximation, using the value φ_{iv_i}, in the computation of x and in the increments to output.

As stated earlier, p is equal to u_i throughout, in the current version of the algorithm. Excursions could be performed by varying p. Also, the investment pattern is assumed to go right up through the period v_i, where the shortfall occurs. Formulas similar to those above could be developed under the assumption that equal-sized investments are completed in each period p through p', where $p' < v_i$, and the accumulated output by the end of period v_i just covers the shortfall. (It can be shown that for a given u_i, v_i and w_i, the total investment $x(p' - p + 1)$ is minimized if $p = p' = u_i$—but this early investment would induce earlier investment demand, which would be more likely to induce an irremediable shortfall.)

2. Computation of the Induced Investment Demand

This section derives the investment demand that is induced by the investment pattern that ameliorates a certain shortfall in a certain industry on a given iteration of the investment algorithm. All notation is the same as in previous sections. Assume that we are trying to meet a shortfall of w_i in industry i in period v_i, by a staggered sequence of investments, each of size x, as described previously. The value of x has been determined in accordance with the preceding subsection.

As stated earlier, if $\eta'_{j\tau it}$ denotes the investment demand on industry j in month τ arising from an investment of size x completed at the beginning of month t, then $\eta'_{j\tau it}$ is given by

$$\eta'_{j\tau it} = \begin{cases} K_{ji}x / l_i & \tau = t - l_i, \ldots, t-1 \\ 0 & \text{for all other } \tau. \end{cases}$$

If an investment of amount x is completed at the beginning of each month t between u_i and v_i, inclusive, the induced investment demand on industry j during period τ is then

$$\eta_{j\tau i} = \sum_{t=u_i}^{v_i} \eta'_{j\tau it} \ .$$

This section derives a closed form for the $\eta_{j\tau i}$.

We assume that K_{ji} is nonzero (i.e., strictly positive); also, we drop the subscript i on most variables. We can then ask: For a given month τ, for which months t does an investment completed at the beginning of month t yield an investment demand at month

τ? The answer: precisely those t such that τ is in the set $\{t-l, t-l+1,\ldots,t-1\}$, i.e., those t such that $1 \le t-\tau \le l$, or equivalently, those t such that $\tau+1 \le t \le \tau+l$. Letting

$$Z(\tau)=\{t \mid \tau+1 \le t \le \tau+l \text{ and } u \le t \le v\},$$

it is evident that

$$\eta_{j\tau i} = \left(\frac{xK_{ji}}{l}\right)|Z(\tau)|.$$

Note that $Z(\tau)$ does not depend on j.

The problem thus reduces to finding the cardinality of the set $Z(\tau)$, as a function of τ. Clearly, if $\tau+l < u$ or $\tau+1 > v$, then $Z(\tau)$ is the null set. Otherwise, let us examine what happens to the intersection of the sets $\{\tau+1, \ldots, \tau+l\}$ and $\{u, u+1, \ldots, v\}$ as τ moves upward from $u-l$. It is instructive to examine separately the cases $l \le (v-u+1)$ and $l > (v-u+1)$. For each case, the bounds on τ and the corresponding cardinality of $Z(\tau)$ are shown in the tables below.

Table III-1. Cardinality of Set $Z(\tau)$ Under Case A: $l \le (v-u+1)$

Condition on τ	Cardinality of Set $Z(\tau)$
$\tau+1 \le u \le \tau+l \le v$	$\tau+l-u+1$
$u \le \tau+1 \le \tau+l \le v$	l
$u \le \tau+1 \le v \le \tau+l$	$v-\tau$

Table III-2. Cardinality of Set $Z(\tau)$ Under Case B: $l > (v-u+1)$

Condition on τ	Cardinality of Set $Z(\tau)$
$\tau+1 \le u \le \tau+l \le v$	$\tau+l-u+1$
$\tau+1 \le u \le v \le \tau+l$	$v-u+1$
$u \le \tau+1 \le v \le \tau+l$	$v-\tau$

By use of the maximum and minimum functions, these two cases can be combined into one. Doing this and also rearranging the conditions on τ to isolate τ yields the results shown in Table III-3.

Table III-3. Cardinality of Set $Z(\tau)$—General Case

Condition on τ	Cardinality of Set $Z(\tau)$
$u-l \le \tau \le \min\{ u-1 , v-l \}-1$	$\tau+l-u+1$
$\min\{ u-1 , v-l \} \le \tau \le \max\{ u-1 , v-l \}$	$\min\{l, v-u+1\}$
$\max\{ u-1 , v-l \}+1 \le \tau \le v-1$	$v-\tau$

For each value of j, in turn, the computer code loops over values of τ, computes

$$\eta_{j\tau i} = \left(\frac{xK_{ji}}{l}\right)|Z(\tau)|,$$

and increments the investment demand $\Delta_{j\tau}^{(m)}$ by $\eta_{j\tau i}$. Note that the limits and values that determine the cardinality of $Z(\tau)$ depend only on i, not j, and are computed outside the loop on j.

IV. THE FORCEMOB COMPUTER PROGRAM

The FORCEMOB computer program implements in a straightforward manner the calculations of the FORCEMOB model described in the preceding chapters. It is written in a relatively portable version of FORTRAN (see Section B.2); there are about 14,500 lines of code. Since what the program does has been discussed rather thoroughly, this chapter, describing the program itself, is relatively brief. Volume II of this paper contains more detail on the program code that relates to the input data. For more detail on other portions of the code, the reader is referred to the code itself.

This chapter begins with a brief description of the inputs, operation, and outputs of the computer program. Then it discusses the program's variables that account for time, and highlights some issues that arise in converting FORCEMOB to other machines and compilers. The chapter concludes with a table that shows the routines of the model.

A. INPUTS, PROGRAM EXECUTION, AND OUTPUTS

1. Inputs and Program Execution

FORCEMOB uses about 25 different types of input data files, although not all of these files will be used in one FORCEMOB run. The files can be grouped into the Run file, the Control Inputs file, and all the other input files. This file organization arises from the following structure:

- Each execution of the FORCEMOB computer program consists of one or more "runs" of the FORCEMOB model.
- Each run is associated with exactly one Control Inputs file.
- Each execution of the program is associated with one Run file, which lists the names of the Control Inputs files for the runs to be executed.
- A Control Inputs file specifies the other input data files to be used, as well as scenario dates, a list of desired output reports, and several key input parameters.[1]

---

[1] The structure of the Control Inputs file is discussed in detail in Volume II, Chapter II.

The other input files can be grouped into main data (or "database") files, optional files and auxiliary files. Volume II of this paper explains this taxonomy and describes all the files. Among the main data files, the Element file is especially noteworthy. It contains the names of the industry sectors and Major End Items—data that affect virtually every part of FORCEMOB. Volume II explains in detail the format of the Run file, the Control Inputs file, and each of the other data files. In total, the input data files can take up several megabytes of disk space; the user should plan for this.

Each run of the model consists of the following steps:

1. Read from the Control Inputs file certain parameter values and the names of the other input data files.
2. Read the other input data files.
3. Start performing the calculations of the Requirements module, the Industry-level module, or both (depending on which have been specified in the Control Inputs file).
4. As appropriate, read additional parameters from the Control Inputs file, read additional data files, and perform additional calculations.
5. From the Control Inputs file, read the list of output reports desired and generate each output report.

To get the program started, the name of the Run file must be supplied to it. There are currently two "sub-versions" of FORCEMOB Version 3.1 operative at IDA, one on a PC (compiled under the Microsoft FORTRAN PowerStation compiler [12]) and one on a DEC VAX (running under VMS). In the PC version the name of the Run file is a command line argument. To run the PC version, the user types (or puts in a batch file)

(name of executable program file) (name of Run file)

If desired, a full path name can be used for the Run file if the length does not exceed 80 characters.

In the VAX version, the program reads the name of the Run file from standard input (the keyboard or a command procedure file) at the beginning of program execution. The READ statement is in the FORTRAN list-directed (*) format, so the name of the Run file must be encased in single quotes. If desired, a full path name can be used if the length does not exceed 80 characters. The command to execute the VAX program version is

RUN (name of executable program file).

This command can be entered from the keyboard or a command procedure file.

The association of the Run file name with the program is one of the features that will need attention in converting FORCEMOB to other machines or FORTRAN compilers; see section B.2.

The execution time of FORCEMOB depends on, among other things, the type of computer, the options being exercised, and the number of output reports being requested. In our experience, Version 3.1 takes from 1 to 5 minutes per run.

2. Outputs

a. History File and Standard Output

Each run of FORCEMOB generates a "history file," which summarizes the run. (One execution of the program will generate a separate history file for each of the runs the program performs.) The history file contains a restatement of (most of) the information in the Control Inputs file, including the names of the data files, scenario dates, and parameter values.[2] It also contains values of certain key summary output measures. (FORCEMOB Version 1.0 generates a history file only if the user requests one. Version 3.1 always generates a history file, as do Versions 2.2 and 3.0.) The history file has the name and directory of the Control Inputs file, and the extension '.HIS'.

In Industry-level module runs, the history file contains a summary output table that shows, for each year of the scenario period, supply, demand by component, imports, and exports. This summary table is also written in a comma-delimited format to a separate little output file, with the same name and directory as the history file, and the extension '.HSC'.

During the course of execution, the FORCEMOB program writes a number of informative messages to standard output. By default, standard output is the terminal, but the user can redirect it to a file if desired. (On the VAX, if the program is executed in batch mode, standard output goes to the .LOG file for the batch job.)

b. Output Reports

The major portion of the FORCEMOB output consists of reports of various results computed by the model. A wide variety of output reports are available, for both

---

2 The Control Inputs file can contain comments, which are not reproduced on the history file.

the Requirements and Industry-level modules. These reports are listed in Table II-4 of Volume II. FORCEMOB generates only those reports that the user has requested on the Control Inputs file. Volume II, Chapter II, explains how to format these requests. Each output report is written to a separate file. FORCEMOB itself does not print these files; the user can print or view them after FORCEMOB finishes executing.

The Version 1.0 "special printed reports" discussed in Chapter IX, section D, of [1] are still available in Version 3.1. These reports include:

- MEI requirements (in dollar or quantity terms)
- Supply and demand (by component), by month or year—separate little table for each industry
- Ranked industry shortfalls
- Materials postprocessor files, which show the components of demand by industry and year or quarter—separate table for each component.

In both FORCEMOB Version 1.0 and the current version, the "quarterly materials postprocessor" report file provides the link between FORCEMOB and the Stockpile Sizing Module [4]. An output of FORCEMOB, this file is input to the SSM. Most of the Version 1.0 screen output reports described in Chapter IX of [1] have been adapted into file output reports. A number of additional output reports, such as pre-scenario demand and TOE, consumption, and threat item usage, have been developed for Version 3.

Most of the output reports are available in two forms. In one form, values on a line are separated by commas, to facilitate reading the file into a spreadsheet. In the second form, values are separated by blank spaces; the file is suitable for printing or for viewing with a text editor.

FORCEMOB Version 1.0 can produce a number of graphical output reports. The code for these is written in a software package that IDA has installed only on the VAX. To enhance code portability, these outputs have been eliminated in Version 3 (if desired, they could be reinstated in a separate postprocessor program). The new graphical user interface for the PC FORCEMOB Version 3.1 can generate graphs from the output reports via the Microsoft Excel spreadsheet software [13].

Caution: A typical output report can take up several hundred kilobytes of disk space. FORCEMOB does not check that sufficient disk space is available before generating an output report. (It might be possible to put such a feature into FORCEMOB, but the computer code to do this would probably vary with the system and

compiler being used.) The user is responsible for ensuring that adequate free disk space exists to hold the output reports requested.

B. SOME SPECIAL ISSUES

1. Program Variables for Treatment of Time

It is not intended or necessary that the general user of FORCEMOB know the details of the FORCEMOB computer code. However, an understanding of FORCEMOB's treatment of time, including the computer code variables used for this, can enhance understanding of the model. Accordingly, this section describes that treatment.

Most of the quantities computed by FORCEMOB are expressed as time streams by month. Calendar years ISYEAR and IEYEAR are read in from the Element database file. They bound all the other times in the FORCEMOB run.[3] The month January of year ISYEAR is referred to as period 1; February of year ISYEAR as period 2, and so forth.[4] In general, calendar month m of calendar year y corresponds to period

$$p=m+(y-\text{ISYEAR})*12.$$

In the FORCEMOB code and output, the phrasing "period p" and the indexing variable IP correspond to the indexing scheme for relative period p as defined above. (When the word "period" is used just before a number or variable name, it means a month, but phrases such as "the scenario period" and "the conflict period" indicate a span of months. The meaning of "period" should be clear from context.)

Table IV-1 shows a "timeline" of the major time-related variables in FORCEMOB. Other than ISYEAR and IEYEAR, the quantities in the "calendar month and year" columns are read in from the Control Inputs file at the start of a FORCEMOB run (as discussed in Volume II). Based on these inputs, the model computes the "period" quantities in the middle column.

---

3 With FORCEMOB's current coding, ISYEAR and IEYEAR cannot cover a time span of more than 15 years. Some reprogramming of FORCEMOB would be necessary to accommodate a longer time span.

4 In this context, the calendar month values are coded as 1 = January, 2 = February, and so forth. Calendar year values have four digits, e.g., 1994.

Table IV-1. Selected FORCEMOB Time-Related Variables

Calendar Month	Calendar Year	Corresponding Period	Definition/Meaning
January	ISYEAR	1	Starting period of overall time accounting
IBEG(1)	IBEG(2)	ISTART	Starting period of scenario
ICBEG(1)	ICBEG(2)	ICONBEG	Start of overall conflict period
IPROF(1,ITHR)	IPROF(2,ITHR)	IBBAT(ITHR)	Start of conflict period in theater ITHR; defined for each theater ITHR
None	None	IEBAT(ITHR)	End of conflict period in theater ITHR; defined as IBBAT(ITHR) + NMON(ITHR) – 1, where NMON(ITHR) is the number of months of conflict in theater ITHR
None	None	ICONEND	End of overall conflict period; defined as maximum over played theaters ITHR of IEBAT(ITHR)
IEND(1)	IEND(2)	ISTOP	Ending period of scenario
December	IEYEAR	NTIME	Ending period of overall time accounting; defined as NTIME = 12*(IEYEAR–ISYEAR+1)

Most quantities that relate to force structure—for example, the inputs in the Force Structure database file (see Volume II) and the computed arrays in the Requirements module code—are indexed by month of conflict period in a given theater. Month IMON of the conflict period in theater ITHR corresponds to period IP = IBBAT(ITHR)+ IMON–1, for IMON from 1 through NMON(ITHR), inclusive. The input NMON(ITHR) is the number of months of conflict in theater ITHR (it is specified on the Control Inputs file). The code moves back and forth between the two indexing systems as appropriate.

If MEI requirements are input (i.e., if an optional MEI requirements file is used), the requirements are given for each month of the conflict period in theater 1, and month IMON corresponds to period IBBAT(1)+IMON–1. In this case, theater 1 is used only as a surrogate theater in which to accumulate the time-phased MEI demands; conflict-related calculations are not performed.

Some quantities are indexed by relative year. Relative year 1 corresponds to calendar year ISYEAR and encompasses periods 1 through 12, relative year 2 corresponds to calendar year ISYEAR + 1 and encompasses periods 13 through 24, and so forth. The indexing variable IYR is often used for relative year.

The following nomenclature is used for certain time spans:

- The "overall calculation period" comprises periods 1 through NTIME, i.e., the total set of periods being considered.
- The "scenario period" or "simulation period" comprises periods ISTART through ISTOP (inclusive).
- The "conflict period" or "overall conflict period" encompasses periods ICONBEG through ICONEND.
- For each theater ITHR, the "conflict period in theater ITHR" encompasses periods IBBAT(ITHR) through IEBAT(ITHR).

The model requires that the overall conflict period lie within the simulation period and that the conflict period for each theater (i.e., each theater that is played) lie within the overall conflict period. All of these time spans must lie within the overall calculation period. If any of these conditions are not satisfied, an error message is printed and the run terminates prematurely.

2. Conversion of the Computer Code to Other Machines

FORCEMOB Version 3.1 is written in FORTRAN 77 and should be able to be converted to other machines fairly easily. An effort has been made to move all special graphic, spreadsheet, and screen-based commands from FORCEMOB itself to separate pre- or postprocessor programs. As stated earlier, FORCEMOB Version 3.1 is now operative on a VAX and a PC. The main non-ANSI code characteristics of FORCEMOB are variable, array, and program segment names of more than six characters and the use of INCLUDE statements. Most FORTRAN compilers allow these extensions. (Symbolic names of more than six characters are *not* necessarily unique in the first six characters.)

In the conversion process, there are two issues that need specific attention. These are:

- The use of date and time functions
- The association of the Run file with the program

The sections below discuss these issues. FORCEMOB is structured so that the program changes necessary to accommodate these factors need be made only to the main program (called Program MAIN, in file MAIN.FOR).

a. Date and Time Functions

FORCEMOB makes use of library functions to get the current date and time. These functions are not only machine-specific but compiler-specific. At the beginning of each run, the main program calls the date and time functions. The current date is encoded into an 11-character string; the time is encoded into another 11-character string. This encoding is performed in the main program. The character strings are stored as character variables in a COMMON block; various subroutines access these variables and print the date and time of the run on the history file and on certain output reports. The date and time are used only for informative purposes and do not affect FORCEMOB's calculation routines. When converting FORCEMOB to other machines, the character strings can simply be left blank. Or, the main program can be modified to make calls to date and time functions provided by the specific compiler in question, and a programmer can develop code to encode the results of these calls into the 11-character strings.

The PC version of FORCEMOB (not the VAX version) includes code (in the main program) that determines the time at the end of program execution and computes the elapsed time. In the conversion process, this code can be omitted or modified as necessary.

b. Association of Run File

As stated in section A.1, in order to run the program, the name of the Run file must be made known to the program in some way. The main program contains an OPEN statement that opens the file with this name and associates it with unit 2. All reads from unit 2 take place in the main program. The Microsoft FORTRAN PowerStation compiler that we have been using for the PC version allows command line arguments, so the PC version lets the name of the Run file be a command line argument. In the conversion process, the programmer can determine the most convenient method of getting the Run file name to the program. In IDA's current VAX version of FORCEMOB, as stated earlier, the program simply reads the name of the Run file from standard input.

C. FORCEMOB PROGRAM SEGMENTS

Table IV-2 lists and briefly describes the program segments of FORCEMOB, including the main program, the subroutines, the one block data routine FILINIT, and the entry IDEBUG. The routines are listed in alphabetical order. The subroutines can be roughly grouped into the following categories:

- Input routines, which read the various input files
- Computational routines, which perform the calculations associated with the model
- Output routines, which generate the output reports
- Miscellaneous utility routines.

For each routine, Table IV-2 shows its category and the module to which the routine belongs.

In the FORCEMOB source code, each routine appears in a separate file; the name of the file is identical or very similar to the routine name. The COMMON block declarations appear in a set of separate files, which are brought into the main source code files by means of INCLUDE statements.

Table IV-2. FORCEMOB Program Segments

	Name	Type	Module	Purpose/Description
1	ACCUM	Computational	Industry-level	Adds up base military, conflict military, and civilian demand, for each industry
2	AGGTAB	Output	Requirements	Generates reports 20 and 24, MEI demands aggregated by budget category
3	ATTCMP	Computational	Requirements	Called by REQC1; computes attrition replacement requirements for a unit
4	CANCEL	Utility	Both	For use with Graphical User Interface, stops program execution at user's request
5	CLR	Utility	Both	Sets all elements of a real array to the value of a specified argument (often zero)
6	CONCMP	Computational	Requirements	Called by REQC1; computes consumption requirements for a unit
7	DEBUG	Utility	Both	Prints values of a real array (for debugging purposes)
8	DLTPRT	Output	Requirements	Prints a summary of pre-scenario conflict military requirements on the history file
9	EUSERS	Output	Requirements	Generates reports 22 and 26, Ranked end users for industry
10	FILINIT	Block Data	Both	Initializes extensions, identification numbers, and default names of input database and optional files
11	FILRST	Utility	Both	Resets input database file names to their default names
12	FILSET	Utility	Both	Stores default names of input database files

continued

Table IV-2. FORCEMOB Program Segments (continued)

	Name	Type	Module	Purpose/Description
13	FORD	Utility	Both	Sorts a data vector in descending order; used to prepare several output reports
14	FORMEI	Output	Requirements	Generates reports 35 and 45, Force units using MEIs
15	ICLR	Utility	Both	Sets all elements of an integer array to the value of a specified argument (often zero)
16	IDEBUG	Utility	Both	Entry in Subroutine DEBUG; prints values of an integer array
17	INVNTRY	Computational	Requirements	Based on inputs, calculates initial allocation of MEI inventory among played theaters
18	INVSHRT	Computational	Industry-level	Performs computations of the investment algorithm
19	LABOR	Computational	Industry-level	(Will be used in future to compute labor requirements)
20	LISTFL	Output	Both	Writes the names of the input data files used in a run (to the history file and various output report files)
21	LOADDAT	Input	Both	Determines input database and optional files that need to be read for the run, calls the routines that read these files
22	MAIN	Main Program	Both	Main program of the FORCEMOB computer program; reads Run file and calls all computational routines
23	MEICAT	Output	Requirements	Generates reports 32 and 42, MEI demands on industry
24	MEITAB1	Output	Requirements	Generates reports 2 and 12, Major End Item dollars report
25	MEITAB2	Output	Requirements	Generates reports 1 and 11, Major End Item units report
26	MILSUP2	Output	Industry-level	Generates report 29, Military supply expansion
27	PMSMSG	Utility	Both	Prints a message when a run stops prematurely
28	PPFILE	Output	Industry-level	Generates reports 5 and 15, Yearly postprocessor file
29	PPFILE2	Output	Industry-level	Generates reports 6 and 16, Quarterly postprocessor file
30	PPTOP	Output	Both	Prints certain summary information about the run at the top of several output files
31	PRSHORT	Output	Industry-level	Generates reports 7 and 17, Ranked shortfall report
32	PSDRPT	Output	Requirements	Generates reports 38 and 48, Pre-scenario demand

continued

Table IV-2. FORCEMOB Program Segments (continued)

	Name	Type	Module	Purpose/Description
33	PSUPXM	Output	Industry-level	Generates reports 8, 18, and 28, Monthly supply expansion file
34	PSUPXY	Output	Industry-level	Generates reports 9 and 19, Yearly supply expansion file
35	RDBAT	Input	Requirements	Reads Control Inputs file data on conflict theater specification, sets up specification
36	RDCFM	Input	Industry-level	Reads Conflict Military Requirements database file
37	RDELEM	Input	Both	Reads Element database file
38	RDFACT	Input	Industry-level	Reads optional factors files (Civilian, Base Military, Import/Export)
39	RDFORCE	Input	Requirements	Reads Force Structure database file
40	RDILMF	Input	Industry-level	Reads Base Military, Civilian, Supply-Side, and Q/K Ratio & Capacity Utilization database files
41	RDINA	Input	Requirements	Reads optional inventory allocation file
42	RDMCF	Input	Industry-level	Reads optional military/civilian fungibility factors file
43	RDMEIRQ	Input	Requirements	Reads optional MEI requirements file
44	RDPDSF	Input	Requirements	Reads Production Process Lead Times and Production Process Matrix database files
45	RDRQMF	Input	Requirements	Reads Cost and MEI Inventories database files
46	RDTHRP	Input	Requirements	Reads Control Inputs file data on theater-related parameters and inventory sharing
47	READ1	Input	Both	Reads Control Inputs file data for basic setup of the run, does that setup
48	READINV	Input	Industry-level	Reads Control Inputs file data on investment-related quantities, computes EOC supply expansion fractions
49	READOUT	Input	Both	Reads Control Inputs file data on output report requests, calls routines to generate output reports
50	REQC1	Computational	Requirements	Converts requirements for TOE, consumption, and threat items to dollar demand for MEIs
51	REQC2	Computational	Requirements	Computes inventory adjustment and conflict military demand on industry
52	REQCAL	Computational	Requirements	Main Requirements module computational routine, calls REQC1 and REQC2
53	REQFOR	Output	Requirements	Generates reports 34 and 44, Requirements per force unit

continued

Table IV-2. FORCEMOB Program Segments (concluded)

	Name	Type	Module	Purpose/Description
54	REQSIM	Computational	Requirements	Computes inventory adjustment and conflict military demand on industry when MEI demands are input
55	RESREQ	Output	Requirements	Generates reports 33 and 43, Industry demands for MEIs
56	RMEIFRAC	Input	Requirements	Reads Control Inputs file data on MEI requirements allocation, if applicable (see Volume II, Chapter II, Options 0b and 1b)
57	SETBASE	Computational	Industry-level	Applies the appropriate (user-specified) base military factors to the base military requirements
58	SHRTFALL	Computational	Industry-level	For each industry and month of simulation, compares supply against demand and determines what shortfalls exist
59	SUPPLYCA	Computational	Industry-level	Determines supply available before investment, considering partial expansion to EOC, dual use, and net imports, as specified by the inputs
60	SUPRPT2	Output	Industry-level	Generates reports 3, 13, and 23, Monthly supply & demand report
61	SUPRPT3	Output	Industry-level	Generates reports 4 and 14, Yearly supply & demand report, and writes summary table on history file
62	TCTOUT	Output	Requirements	Generates reports 21 and 25, TOE, consumption, and threat items required
63	TERMFL	Utility	Both	Writes file OKAY.FLG if program terminated normally, or file ABORT.FLG if program terminated prematurely. Graphical User Interface reads these files.
64	TOECOM	Output	Requirements	Generates reports 36 and 46, TOE composition for MEIs
65	TOEMEI	Output	Requirements	Generates reports 37 and 47, TOE-MEI correspondence
66	TRIMSUP	Computational	Industry-level	Performs computations of supply reduction algorithm
67	TXOUT	Output	Industry-level	Generates report 30, Supply-side spreadsheet-ready output
68	UNITT	Output	Requirements	Generates reports 31 and 41, Force unit delivery profiles
69	UNTCMP	Computational	Requirements	Called by REQC1; computes start-up requirements for a unit
70	WRTCFM	Output	Requirements	Writes out conflict military requirements file

V. FORCEMOB VERSION 3.2

This chapter describes the differences between FORCEMOB Version 3.2 and Version 3.1. There are two variants of Version 3.2. Version 3.2a, although an interim version, was used in a major study (the 1995 National Defense Stockpile Study), and thus some description of it is in order. Version 3.2b is the most current version of FORCEMOB as of this writing. This chapter focuses on Version 3.2b; a section at the end discusses Version 3.2a.

A. METHODOLOGICAL CHANGE

Version 3.2 adds to the FORCEMOB model a new option, which can be called the early MEI exclusion option. This option affects the computation of demand on industry that is induced by conflict-related Major End Item demand that occurs early enough in the scenario to induce pre-scenario industry demand, as discussed in Chapter II, Section A.4.c.[1] The option allows such MEI demand to induce *no* demand on industry. For example, if there is demand for a certain MEI two years into the scenario and the production lead time for that MEI is four years, then none of that MEI demand will induce industry demand—the MEI demand goes unsatisfied because it cannot be produced on time.

A more precise description of the option's operation is as follows. If the option is exercised and all of the following conditions apply:

- the Requirements module generates demand for some MEI in month t of the scenario period
- the production lead time for that MEI is m months
- t is strictly less than m (i.e., the demand occurs "early" in the scenario, so that the lead time extends back before the beginning of the scenario)

1 Recall that such demand occurs when the time period of the demand is earlier in the scenario than the lead time of the MEI, so that the lead time extends back before the start of the scenario period.

then *none* of the demand for that MEI in month t is assumed to induce demand on industry. In contrast, if the option is not exercised, fraction (t/m) of that MEI demand induces demand on industry, as described in Chapter II, Section A.4.c.

B. COMPUTER PROGRAM CHANGES

To implement this new option, several changes have been made to the computer program: a new optional input file, an additional output report, and some changes to the computer code.

1. Changes to Input Files

The early MEI exclusion option is invoked by use of an optional input data file, the Early MEI Exclusion file. This file consists of two lines: the first line can contain any comment the user wishes, and the second line contains a single (integer) numerical value (read in list-directed format and assigned to a program variable named IEXEMD). If this value is zero, then the option is not exercised and the program proceeds as it has in the past. If the value is unity, or any nonzero value, then the program does exercise the option and does not allow early MEI demand to induce industry demand. Of course, if this optional file is not requested, the option is not exercised.

An Optional Early MEI Exclusion file can be requested on the Control Inputs file in the same manner as the other optional input files. Its identification number is 7. The file itself should have the extension '.EME' and should reside in the data file directory for the run. The file, and the whole set of computations relating to the implementation of the option, are considered to be part of the Requirements module of the FORCEMOB model.

Note that all the other FORCEMOB input database and optional files, the Run file, and the auxiliary MEI mapping file, remain the same as described in Volume II. The Debugging Flags file for Version 3.2b is the same as described in Volume II; for Version 3.2a, it is different, as discussed in Section C, below. The Control Inputs file for Version 3.2b has the following differences from that of Version 3.1 (and 3.2a): first, the section where the optional files are specified now allows a listing for Optional File 7; second, the output report request section allows report 49 (described below) to be selected.

2. Changes to Output Reports

An additional output report, identified as report 49 with file extension '.MNC', shows, for each MEI, the MEI demand that does not induce industry demand, the MEI

demand that does induce industry demand, and the total. Values are given both in dollars and in numbers of items. The output report is comma delimited to facilitate transfer to a spreadsheet. (This report is available even if the early MEI exclusion option is not chosen. In this case, it shows the MEI demand that induces pre-scenario industry demand.) This report can be requested on the Control Inputs file in the same manner as the other output reports, as discussed in Volume II, Chapter II.

In addition, if the option is exercised, the Pre-Scenario Demand reports, reports 38 and 48, show, for each industry, the industry demand that would have been induced by the early MEI demands, according to the Production Process matrix. The industry demands that are induced by the later MEI demands, and the totals, are also shown.

3. Changes to the Computer Code

Code (Subroutine RDEME and some changes to Subroutine LOADDAT) has been added to read and process the optional input file (Optional File 7) that specifies the early MEI exclusion indicator. Subroutine RMEIDNC has been added to generate output report 49; a call to RMEIDNC has been added to Subroutine READOUT.

Subroutine REQC2 of Version 3.1 has been broken up. In Version 3.2, Subroutine REQC2 computes the effect of inventory share groups and the final MEI demand net of inventory, including early MEI demand. New Subroutines REQC3A and REQC3B compute the demand on industry under the traditional methodology and the new option, respectively.

Some changes to Subroutine REQSIM have also been made, to handle the case where MEI demands are input directly and the early MEI exclusion option is invoked. (These changes have been implemented in Version 3.2b only.)

FORCEMOB Versions 3.2a and 3.2b have been prepared for the PC; VAX versions do not currently exist. The programs could be converted to the VAX (or other computers) as described in Chapter IV.

C. FORCEMOB VERSION 3.2a

FORCEMOB Version 3.2a is like 3.2b, except the indicator to invoke the early MEI exclusion option is given by the variable IFLAG2 on the Debugging Flags file (see Volume II). If a Debugging Flags file exists in the directory from which the program is

being executed, and if the variable IFLAG2 in this file has a value of 1, then the option is exercised. Otherwise, it is not exercised.

Also, Version 3.2a does not have an output report identified by the number 49. A similar report is produced, but is not handled in the same way as the other FORCEMOB output reports. The differences are:

- The report is generated only if the early MEI exclusion option is exercised.
- The report resides in the directory from which the program is being executed.

In addition, Version 3.2a requires that MEI requirements be computed via a Force Structure file (as opposed to being input) if the early MEI exclusion option is to be exercised.

Appendix

LINEAR PROGRAMMING FORMULATION OF THE FORCEMOB INVESTMENT PROBLEM

CONTENTS

Appendix

LINEAR PROGRAMMING FORMULATION OF THE FORCEMOB INVESTMENT PROBLEM

A. INTRODUCTION AND OVERVIEW

For each month within a user-specified scenario period, FORCEMOB determines the industrial demands arising from conflict military requirements and adds these to input base military and civilian demands to determine a total demand. It calculates an initial level of available supply and compares the supply with the demand to determine whether there are any shortfalls. Supply surpluses that occur early in the scenario can be carried forward to satisfy later demands, but the reverse is not true: demands must be satisfied at the time they occur; otherwise there is a shortfall. If there are shortfalls in some industries, investments can be made in those industries to increase supply and reduce or, hopefully, eliminate the shortfalls.

A tradeoff problem can arise, however, because investments, while increasing supply, might also increase demand in other industries. The process of building plant capacity for one industry involves purchases from other industries, and thus investment in an industry puts a demand on such "feeder" industries. A further complication is time: the increased output comes *after* an investment is completed, while the induced investment demand occurs during the process of investment. We wish to structure the investment such that these induced investment demands are not so large as to create shortfalls in the feeder industries. FORCEMOB contains a heuristic algorithm that tries to do this—but this algorithm is not guaranteed to find a feasible investment pattern, even if one exists.

Under certain assumptions about the investment output and demand structure, this problem—how to invest to ameliorate shortfalls while not creating additional shortfalls

from investment demand—can be formulated as a linear programming problem (LP).[1] This appendix develops several such formulations, varying slightly in their assumptions. Standard LP solution procedures, such as the simplex method, can then identify a feasible investment pattern if one exists, or show that no such pattern exists. While it probably is not practical to embed an LP solution algorithm in the FORCEMOB computer code itself, LP testing can be done as an off-line procedure, to benchmark the heuristic procedure in FORCEMOB.

This appendix is structured as follows. First, we develop a relatively simple, straightforward LP formulation of the FORCEMOB investment problem, clarifying our assumptions and defining appropriate notation as we go. Then, we discuss some ways of reducing the number of variables and constraints in this formulation. After that, we present an additional LP formulation that explicitly considers labor constraints. Finally, we develop an LP formulation that allows some shortfalls in supply but seeks to determine how much of the shortfall can be ameliorated by investment.

B. THE FIRST FORMULATION

This formulation characterizes investments by two features: (1) the industry in which investment is made, and (2) the time that the investment is completed and the resultant additional productive capacity is in place. The decision variables correspond to industry/completion time pairs, and represent the amounts to invest. Before discussing the decision variables further, we develop notation for the appropriate indices and constants.

1. Initial Notation

a. Indices

i, j = indices for industry (sector)

t, u, τ = indices for time period (e.g., month), within the overall scenario period. Where it causes no confusion, the word "period" will be used instead of "time period."

---

1 For information on the structure and solution of linear programming problems, see any text on linear programming, such as Bradley, Hax, and Magnanti [14].

b. Constants

The following entities are constants, or parameters, within the LP formulation. Some of them correspond to inputs to FORCEMOB, others are computed quantities within FORCEMOB.

I = number of industries (sectors); the industries are indexed by i (or j) = $1,\ldots,I$.

T = the number of time periods (e.g., months) in the overall scenario period. In FORCEMOB, the time periods are months, but from the standpoint of the LP formulation, this is not necessary. In fact, the time periods need not be of equal length, as long as appropriate data can be developed. The time periods are indexed as t (or u or τ) $=1,\ldots,T$ (in the computer code, IP from ISTART through ISTOP).

c'_{it} = pre-investment supply in industry i in period t. This encompasses domestic production, adjusted for expanded capacity utilization as desired, plus net imports, but does not consider additional supply created by investment. The quantity c'_{it} corresponds to the *maximum* amount that can be supplied (under the given capacity utilization), the actual amount supplied might be somewhat smaller in certain time periods (see Chapter II, section B.6). We assume that this value is nonnegative. One generally thinks of supply values as measured in dollars or some multiple thereof, but other appropriate units can be used.

d'_{it} = pre-investment demand on industry i in period t. This value encompasses civilian, base military, and conflict military demand, but not investment demand, and is assumed to be nonnegative (to model a negative demand, increase the corresponding pre-investment supply value). Generally measured in dollars or some multiple thereof, but other appropriate units can be used if they are consistent with the units of supply.

h'_{i0} = supply inventory level in industry i at the beginning of the scenario period. The current version of FORCEMOB assumes that there is no initial inventory, but this LP formulation can handle it. Measured in units consistent with supply (generally, dollars) and assumed to be nonnegative.

$v_{itj\tau}$ = amount of investment demand on industry i in period t that is induced by an investment of one dollar (or some multiple thereof) in industry j that is completed in period τ. The value of $v_{itj\tau}$ can reflect the investment distribution (capital coefficients) and the investment lead time associated with industry j. It might also depend on the length of the time periods. The list of industry sectors (i=$1,\ldots,I$) should span the economy, so that any industry in which investment demand can be induced is encompassed by the list of industries considered, at some level of aggregation. Note that $v_{iti\tau}$ is allowed to have a

strictly positive value, the interpretation being that an industry might have to use its own products to expand its productive capacity.

$\xi_{it\tau}$ = amount of additional supply in industry i in period t that is induced by an investment of one dollar (in industry i) that is completed in period τ. This value can take into account the capital/output ratio for industry i, and might also depend on the length of the time periods. Although the time subscripts allow full generality, it is expected that $\xi_{it\tau}$ would be zero for $t < \tau$.

c. Decision Variables

The decision variables are the investment amounts.

$x_{j\tau}$ = number of dollars of to invest in industry j in a manner such that the investment is completed in period τ, defined for each industry j and time period τ where such investment is feasible (see the discussion in section 2.a, below). The $x_{j\tau}$ are constrained to be nonnegative.

The process of building new capacity in a given industry, industry j, involves the procurement of output from a number of different "feeder" industries over some period of time and using this output to build factories, machinery, and the like, that can produce the products of industry j. In the context of the LP formulation, we assume that this investment process is completely characterized by the time-dependent set of coefficients $v_{itj\tau}$ and $\xi_{it\tau}$. The investment assumptions within the FORCEMOB model itself, as discussed in Chapter III, correspond to a special case of these coefficients.

2. Development of the LP

a. Development of Constraints and Additional Variables

For each i and t, the total investment demand in industry i at time t induced by the investment pattern $\{x_{j\tau}\}$ is given by

$$\Delta_{it} = \sum_{j=1}^{I} \sum_{\tau=1}^{T} v_{itj\tau} x_{j\tau} \ .$$

The total demand, base plus investment, in industry i at time period t, is then

$$d_{it} = d'_{it} + \Delta_{it} \ .$$

Similarly, the total induced supply in industry i at time t is

$$\Gamma_{it} = \sum_{\tau=1}^{T} \xi_{it\tau} x_{i\tau} \quad ,$$

and the total supply, pre-existing plus newly-induced, in industry i at time t, is

$$c_{it} = c'_{it} + \Gamma_{it} \quad .$$

In the first time period, let us consider the quantity

$$h_{i1} = h'_{i0} + c_{i1} - d_{i1} \quad .$$

A negative value of h_{i1} indicates a shortfall in industry i at the end of period 1: supply plus initial inventory was insufficient to meet demand. A feasible solution to the LP requires that there be no shortfall, so we put the constraints

$$h_{i1} \geq 0, \qquad i = 1, \ldots I$$

into the LP. Given that h_{i1} is nonnegative, it represents a surplus (possibly zero) supply inventory, which we assume can be used to help offset future demands, no matter how far into the future they may be. Accordingly, for time periods t =2,...,T, we define

$$h_{it} = h_{i,t-1} + c_{it} - d_{it} \quad ,$$

the shortfall or surplus at the end of period t, and put in the constraints

$$h_{it} \geq 0 \quad i = 1, \ldots, I, \quad t = 2, \ldots, T,$$

to ensure that the values h_{it} are surpluses, not shortfalls. The quantity h_{it} can be thought of as the accumulated net inventory at the end of period t. (There is an implicit assumption that all the supply in period t can be used to satisfy demand in period t; we do not distinguish the precise timing of supply and demand within a time period.)

It is possible that for certain industry/time period combinations j and τ, it simply is not possible to invest in industry j and complete the investment at time τ. In FORCEMOB, for example, no investment in industry j can be completed until after l_j months into the scenario, where l_j is the investment lead time for industry j (see Chapter III, section A). We can add explicit constraints $x_{jt} = 0$ for such j and τ combinations. Alternatively, we can simply not include the variables x_{jt} in the LP formulation. The latter approach is probably the one to take when actually setting up the LP with numerical values for machine solution.

b. Objective Function

The form of the objective function is not too important, as we are primarily trying to determine whether any feasible investment pattern exists (this can be done via a Phase I simplex procedure; also see section E). One reasonable objective function is to minimize the total investment, i.e.,

$$\text{minimize} \sum_{j=1}^{I} \sum_{\tau=1}^{T} x_{j\tau} \quad .$$

3. Formal Problem Statement

With the constants, variables, and constraints as developed in the preceding sections, we can formally state the FORCEMOB investment problem as the following LP:

$$\text{minimize} \sum_{j=1}^{I} \sum_{\tau=1}^{T} x_{j\tau}$$

subject to

$$\begin{aligned}
&\Delta_{it} = \sum_{j=1}^{I} \sum_{\tau=1}^{T} v_{itj\tau} x_{j\tau} \\
&d_{it} = d'_{it} + \Delta_{it} \\
&\Gamma_{it} = \sum_{\tau=1}^{T} \xi_{it\tau} x_{i\tau} \\
&c_{it} = c'_{it} + \Gamma_{it} \\
&h_{i1} = h'_{i0} + c_{i1} - d_{i1} \\
&h_{it} = h_{i,t-1} + c_{it} - d_{it} \quad , \quad t = 2,\ldots,T \\
&h_{i1} \geq 0, \quad i = 1,\ldots,I \\
&h_{it} \geq 0, \quad i = 1,\ldots,I, \quad t = 2,\ldots,T. \\
&x_{j\tau} \geq 0 \\
&x_{j\tau} = 0 \quad \text{for specified pairs } (j,\tau),
\end{aligned}$$

where i and j range from 1 through I and t and τ range from 1 through T, unless otherwise specified.

As currently formulated, the LP has $6IT$ constraints, not counting the nonnegativity constraints and the constraints $x_{j\tau} = 0$. In typical FORCEMOB data sets, I and T have both had values on the order of 200 or 300, so the LP would have on the order of 300,000 constraints! This is very large for commercial LP packages. However, it is easy to condense the problem, as shown in the following section.

C. CONDENSING THE NUMBER OF VARIABLES AND CONSTRAINTS

The number of constraints and variables in the problem can be reduced greatly by substituting for some of the intermediate variables. First note that in a feasible solution,

$$h_{it} = h'_{i0} + \sum_{u=1}^{t} c_{iu} - \sum_{u=1}^{t} d_{iu}\text{, for all } i \text{ and } t.$$

Let us define the additional constants

$$C'_{it} = \sum_{u=1}^{t} c'_{iu} \quad \text{and}$$

$$D'_{it} = \sum_{u=1}^{t} d'_{iu}$$

Then substituting for c_{iu} and d_{iu} in the equation for h_{it} yields

$$h_{it} = (h'_{i0} + C'_{it} - D'_{it}) + \sum_{u=1}^{t}\sum_{\tau=1}^{T} \xi_{iu\tau} x_{i\tau} - \sum_{u=1}^{t}\sum_{j=1}^{I}\sum_{\tau=1}^{T} v_{iuj\tau} x_{j\tau}$$

The "no shortfalls" constraint $h_{it} \geq 0$ is thus equivalent to a constraint involving a linear combination of the decision variables $x_{j\tau}$. The FORCEMOB investment problem can then be formulated as the LP

$$\text{minimize} \sum_{j=1}^{I}\sum_{\tau=1}^{T} x_{j\tau}$$

subject to

$$\sum_{u=1}^{t}\sum_{j=1}^{I}\sum_{\tau=1}^{T} v_{iuj\tau} x_{j\tau} - \sum_{u=1}^{t}\sum_{\tau=1}^{T} \xi_{iu\tau} x_{i\tau} \leq (h'_{i0} + C'_{it} - D'_{it}) \quad i = 1,\ldots,I, \quad t = 1,\ldots,T.$$

$$x_{j\tau} \geq 0$$

$$x_{j\tau} = 0 \quad \text{for specified pairs } (j,\tau).$$

This LP only has IT constraints, plus the nonnegativity constraints and the special constraints $x_{j\tau} = 0$. This is within the capability of commercial LP packages. The main constraints have the interpretation that at any time period, the *net* demand created by the investment process must not exceed the net pre-investment supply.

D. AN LP FORMULATION THAT CONSIDERS LABOR CONSTRAINTS

1. Introduction and Additional Notation

The current version of the FORCEMOB model does not explicitly consider the labor needed to produce the additional output that the investment makes possible. It assumes that all necessary labor is available when needed. Consideration of labor constraints might be modeled in future versions of FORCEMOB. And labor constraints can be added to the LP formulation of the investment problem, as shown in this section.

Heretofore, we have often used the word "supply" to mean supply *capacity*, the *maximum* amount that can be supplied. To model labor constraints, however, we must distinguish supply capacity from the amount actually supplied. We therefore establish the following additional decision variables:

s_{it} = amount of output actually supplied (produced) in industry i in time period t, defined for i=1,..., I and t=1,..., T.

Let us introduce two new constants:

λ_i = amount of labor necessary to produce one unit (e.g., dollar's worth) of output in industry i. Defined for i=1,..., I. Units for λ_i might be dollars or worker-hours per unit of output.

L_t = amount of labor available in time period t. Units might be dollars or worker-hours, consistent with λ_i. Defined for t=1,..., T. In this formulation, L_t is a function of time but not of industry. This is consistent with the assumption that labor is completely fungible between different industries. Some different assumptions will be explored later on.

2. Development of the LP

With the additional notation defined above, and keeping all other notation defined in previous sections, we can develop the LP formulation of the investment problem with

labor constraints. As before, for each i and t, the total investment demand in industry i at time t induced by the investment pattern $\{x_{j\tau}\}$ is given by

$$\Delta_{it} = \sum_{j=1}^{I} \sum_{\tau=1}^{T} v_{itj\tau} x_{j\tau} \quad ,$$

the total demand is given by

$$d_{it} = d'_{it} + \Delta_{it} \quad ,$$

the newly-induced supply *capacity* is

$$\Gamma_{it} = \sum_{\tau=1}^{T} \xi_{it\tau} x_{i\tau} \quad ,$$

and the total supply capacity is

$$c_{it} = c'_{it} + \Gamma_{it} \, .$$

We require that s_{it}, the amount actually produced in industry i in time period t, not exceed the capacity c_{it}, i.e., the constraints

$$s_{it} \le c_{it} \quad i = 1,\ldots,I, \quad t = 1,\ldots,T$$

are part of the LP formulation. Of course, the s_{it} are constrained to be nonnegative. The inventory h_{i1} at the end of period 1 is now defined by

$$h_{i1} = h'_{i0} + s_{i1} - d_{i1} \quad ,$$

and to avoid shortfalls, h_{i1} is constrained to be nonnegative. For time periods t=2,..., T, the inventory definition equation becomes

$$h_{it} = h_{i,t-1} + s_{it} - d_{it} \quad ,$$

and as before, the constraints

$$h_{it} \ge 0 \quad i = 1,\ldots,I, \quad t = 2,\ldots,T$$

appear to ensure that the values h_{it} are surpluses, not shortfalls. If the supply capacity is strictly greater than the demand for a number of time periods, there might be many feasible production patterns $\{s_{it}\}$ for a given set of capacities $\{c_{it}\}$. To avoid excess production and labor usage, the final remaining inventory h_{iT} can be constrained to equal zero.

The labor constraints are as follows. The total amount of labor used to produce the output of time period t is $\sum_{i=1}^{I} \lambda_i s_{it}$, which should not exceed the labor amount, L_t, that is available at time t.

As before, we add the constraints $x_{j\tau} = 0$ for those industry/time period combinations j and τ where investment is not possible. The objective function

$$\text{minimize} \sum_{j=1}^{I} \sum_{\tau=1}^{T} x_{j\tau} \quad ,$$

to minimize the total amount of investment, is still reasonable.

3. Condensation of Constraints

As before, the number of constraints and variables in the problem can be reduced by substituting for the intermediate variables. In a feasible solution,

$$h_{it} = h'_{i0} + \sum_{u=1}^{t} s_{iu} - \sum_{u=1}^{t} d_{iu} \text{, for all i and t,}$$

and d_{iu} can be replaced by a linear (affine) function of the $x_{j\tau}$, yielding

$$h_{it} = (h'_{i0} - D'_{it}) + \sum_{u=1}^{t} s_{iu} - \sum_{u=1}^{t} \sum_{j=1}^{I} \sum_{\tau=1}^{T} v_{iuj\tau} x_{j\tau} \quad .$$

This definition can be substituted in the constraints $h_{it} \geq 0$ and $h_{iT}=0$. The supply capacity c_{it} can be similarly replaced in the constraints $s_{it} \leq c_{it}$. (The s_{it} themselves are decision variables, and cannot be simplified via substitution.)

4. Formal Problem Statement

With the constants, variables, and constraints as developed in the preceding sections, we can formally state the FORCEMOB investment problem with labor constraints as the following LP.

$$\text{Minimize} \sum_{j=1}^{I}\sum_{\tau=1}^{T} x_{j\tau}$$

subject to

$$s_{it} - \sum_{\tau=1}^{T} \xi_{it\tau} x_{i\tau} \le c'_{it} \qquad i = 1,\ldots,I, \quad t = 1,\ldots,T \qquad \text{(supply doesn't exceed capacity)}$$

$$\sum_{u=1}^{t}\sum_{j=1}^{I}\sum_{\tau=1}^{T} v_{iuj\tau} x_{j\tau} - \sum_{u=1}^{t} s_{iu} \le (h'_{i0} - D'_{it}) \qquad i = 1,\ldots,I, \quad t = 1,\ldots,T-1 \qquad \text{(no shortfalls)}$$

$$\sum_{u=1}^{T}\sum_{j=1}^{I}\sum_{\tau=1}^{T} v_{iuj\tau} x_{j\tau} - \sum_{u=1}^{T} s_{iu} = (h'_{i0} - D'_{iT}) \qquad i = 1,\ldots,I \qquad \text{(final inventory is zero)}$$

$$\sum_{i=1}^{I} \lambda_i s_{it} \le L_t \qquad t = 1,\ldots,T \qquad \text{(labor constraints)}$$

$$s_{it} \ge 0$$

$$x_{j\tau} \ge 0$$

$$x_{j\tau} = 0 \text{ for specified pairs } (j,\tau)$$

This LP has $2IT + T$ constraints, plus the nonnegativity constraints and the special constraints $x_{j\tau} = 0$.

5. Alternative Ways to Model Labor

The above formulation has assumed complete fungibility of labor across industries, with one pool of labor available at each time period. An alternative model could posit a separate pool of labor, L_{it}, for each industry at each time period. The interpretation would be that each industry has its own specialized workers that can work only in that industry. Then for each time period and industry, the total labor used could not exceed the labor available, i.e., the labor constraints would appear as

$$\lambda_i s_{it} \le L_{it}, \quad i = 1,\ldots,I, \quad t = 1,\ldots,T.$$

This amounts to an upper-bounding constraint on s_{it}.

A more realistic assumption might be that each industry needs some specialized labor and some fungible labor. Or, a separate index on occupation could be established. There would be a pool of available labor for each occupation and time period, and each

industry would require a certain amount of labor from each occupation per unit of output produced. Appropriate constraints could be worked into the LP format. Of course, actual solution of such an LP would be contingent on the development of the necessary data values.

E. HOW MUCH SHORTFALL CAN BE REDRESSED THROUGH INVESTMENT?

It is certainly possible that there does not exist a pattern of investment that can redress all shortfalls in supply. In this case, one logical next question is how much of the shortfall *can* be redressed? This question can be investigated via an LP formulation, the solution of which provides a phase I procedure for the LP formulated in sections B and C. For simplicity, we work with that formulation, rather than the labor constraints version.

1. Assumptions and Derivation

Let all notation be the same as in sections B and C. There, we treated the non-investment demands as givens, expressed by the constants, d'_{it}, which were required to be satisfied. However, suppose that it were possible simply to not address some of this demand—just let it go unsatisfied—and redress the remainder of the demand by investment or by existing supply. Supply (initial plus investment-induced) would be required to cover this "remainder of the demand" plus any investment demand.

The LP formulation involves establishing additional decision variables, y_{it}, that represent the amount of non-investment demand in industry i in period t that is to be redressed (for each i and t). (The investment amounts $x_{j\tau}$ remain decision variables also.) Each variable y_{it} can assume values between zero and d'_{it}; these upper-bounding constraints will appear in the formulation. If we follow the derivation in sections B and C but "pretend" that only y_{it}, rather than the full amount of non-investment demand, is to be satisfied, then the other constraints of the LP formulation will look just like those of sections B and C, but with the total non-investment demands d'_{it} replaced with y_{it}.

We want to find an investment pattern that will redress as much of the demand as possible. Accordingly, for the objective function, we can use

$$\text{maximize} \quad \sum_{i=1}^{I}\sum_{t=1}^{T} y_{it} .$$

Along with the upper bounding constraints

$$y_{it} \le d'_{it},$$

this objective function will have the effect of minimizing the total of the demands $(d'_{it} - y_{it})$ that we do not consider. These amounts correspond to demands that must be met in ways other than investment, for example, by increased imports or civilian austerity. If there is a solution where $y_{it} = d'_{it}$, i.e., if it is possible to meet all the demand via investment, then this LP will find that solution. In this case, the result becomes an initial feasible solution for the LP of sections B and C.

2. LP Formulation

With the above remarks in mind (and observing the formulation in section C), we can formulate the problem of finding out how much shortfall can be redressed as

$$\text{maximize} \quad \sum_{i=1}^{I}\sum_{t=1}^{T} y_{it}$$

subject to

$$\sum_{u=1}^{t}\sum_{j=1}^{I}\sum_{\tau=1}^{T} v_{iuj\tau}x_{j\tau} - \sum_{u=1}^{t}\sum_{\tau=1}^{T} \xi_{iu\tau}x_{i\tau} + \sum_{u=1}^{t} y_{iu} \le (h'_{i0} + C'_{it}) \qquad i = 1,\ldots,I, \quad t = 1,\ldots,T.$$

$$0 \le y_{it} \le d'_{it}$$

$$x_{j\tau} \ge 0$$

$$x_{j\tau} = 0 \quad \text{for specified pairs } (j,\tau)$$

This LP has IT constraints, plus the nonnegativity and upper bounding constraints and the special constraints $x_{j\tau} = 0$. Note that an initial feasible solution to *this* LP is evident: set all y_{it} and $x_{j\tau}$ to zero (make no investments and let all demand go unsatisfied).

3. Refinements to the Formulation

A number of refinements and additions to this LP formulation are possible. Labor constraints could be added to it, in accordance with the formulation of section D. The objective function could be some kind of weighted sum of the y_{it}, instead of a simple sum (it is unclear exactly which weights would be appropriate).

Another refinement to the LP involves separating the components of non-investment demand. These components include conflict military demand, base military

demand, civilian demand, and, possibly, exports. Different decision variables could be introduced to represent the amount of each component to be satisfied. Under current FORCEMOB assumptions, the conflict military industry demand is itself a linear function of Major End Item demand. One could go back to the MEI level, introducing decision variables that represent the amount of MEI demand, by MEI type and time period, to attempt to satisfy. These variables would be constrained to lie between zero and the specified requirement for the MEI. The corresponding industry demands would become intermediate variables in the LP, contributing to the total industry demand values.

REFERENCES

[1] Flythe, Mary Catherine, Timothy T. Fitzpatrick, Tara Santmire, Eleanor L. Schwartz, and Richard H. White, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume II, *Users Guide.* IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.

[2] Thomason, James S., Eleanor Schwartz, An-Jen Tai, Sean Barnett, and David Graham, *IDA Analyses for the Naval Logistics 2001 Wargame '94.* IDA Paper P-2954, Institute for Defense Analyses, Alexandria, VA, May 1994.

[3] Thomason, James S., D. Sean Barnett, Tara E. Santmire, Eleanor L. Schwartz, An-Jen Tai, and Richard H. White, *National Defense Stockpile Program Phase II: Assessments and Methodological Enhancements.* IDA Paper P-2885, Institute for Defense Analyses, Alexandria, VA, October 1993.

[4] Santmire, Tara E., *Stockpile Sizing Module Documentation for Version 4.0.* Volume I, *Theoretical Foundations and Users Guide*, Volume II, *SSM Inputs*, and Volume III, *SSM Outputs.* IDA Paper P-2867 (draft) , Institute for Defense Analyses, Alexandria, VA, June 1994.

[5] White, Richard H., project leader, Eleanor L. Schwartz, Mary Catherine Flythe, and Timothy T. Fitzpatrick, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume III, *Programmers Guide.* IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.

[6] *Standard Industrial Classification Manual 1972.* Executive Office of the President, Office of Management and Budget, 1972.

[7] Schwartz, Eleanor L., and Richard H. White, *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume IV, *Data Preparation Guide.* IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.

[8] White, Richard H., *Documentation for the Forces Mobilization Model (FORCEMOB)*, Volume I, *Theoretical Foundations.* IDA Paper P-2716, Institute for Defense Analyses, Alexandria, VA, July 1992.

[9] Thomason, James S., *The JIMPP Requirements Module: Concept and Data Base Development Plan.* Working Paper #88-5, Task T-I6-314, Institute for Defense Analyses, Alexandria, VA, September 1988.

[10] Frazier, Thomas P., Carol G. Campbell, and Richard T. Cheslow, *The Revised Defense Translator.* IDA Paper P-2141, Institute for Defense Analyses, Alexandria, VA, October 1989.

[11] Miller, Ronald E., and Peter D. Blair, *Input/Output Analysis: Foundations and Extensions.* Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[12] *Microsoft FORTRAN PowerStation Professional Development System*, Version 1.0. Microsoft Corporation, Redmond, WA, 1992.

[13] *Microsoft Excel,* Version 5.0. Microsoft Corporation, Redmond, WA, 1993.

[14] Bradley, Stephen P., Arnoldo C. Hax, and Thomas L. Magnanti, *Applied Mathematical Programming.* Addison-Wesley, Reading, MA, 1977.

GLOSSARY

ANSI	American National Standards Institute
DEC	Digital Equipment Corporation
DoD	Department of Defense
EOC	Emergency Operating Capacity
FEMA	Federal Emergency Management Agency
FM	Forces Mobilization Model (FORCEMOB)
FMS	Forms Management System
FORCEMOB	Forces Mobilization Model
GDP	Gross Domestic Product
IDA	Institute for Defense Analyses
ILM	Industry-level Module (of FORCEMOB)
JIMPP	Joint Industrial Mobilization Planning Process
LP	Linear Programming problem
MEI	Major End Item
NDS	National Defense Stockpile
OSD	Office of the Secretary of Defense
PC	Personal Computer
Q/K Ratio	Capital/Output Ratio
REQMOD	Requirements Module (of FORCEMOB)
SIC	Standard Industrial Code
SSM	Stockpile Sizing Module (part of JIMPP)
TOE	Table of Organization and Equipment
VLM	Vendor Level Model (part of JIMPP)
VMS	Virtual Memory System

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
January 1996

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Documentation of the Forces Mobilization Model (FORCEMOB) Versions 3.1 and 3.2, Volume I: General Description

5. FUNDING NUMBERS
C-DASW01-94-C-0054
TA- T-AO6-656

6. AUTHOR(S)
Eleanor L. Schwartz, An-Jen Tai, Richard H. White, James S. Thomason

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2953

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of the Assistant Secretary of Defense (Economic Security)
The Pentagon, Room 2A318
Washington, DC 20301

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This paper provides documentation of Versions 3.1 and 3.2 of the Institute for Defense Analyses' Forces Mobilization Model (FORCEMOB). FORCEMOB is a quantitative model of the effect of an extraordinary military demand, such as a conflict or reconstitution, on the industrial base of the United States. It computes requirements for weapons and consumables that arise from a user-specified mobilization and conflict scenario that can last several years and can involve all Services and up to four theaters. Alternatively, time-phased military requirements can be input. These requirements are converted to demands on industry, which are added to any other military and civilian industrial demand. The model then compares total demand with industry supply and notes shortfalls, if any. The process of capacity expansion and investment to redress shortfalls is modeled. FORCEMOB accounts for the impact of time: most input data can vary by year, and lead times for production and investment are considered. The model contains many adjustment parameters that can be varied to conduct sensitivity analyses easily. FORCEMOB has been used in assessments of the National Defense Stockpile and in other studies of the industrial base.

14. SUBJECT TERMS
Computer Documentation, Modeling, Wargame, Economic Forecasting, National Defense Stockpile, FORCEMOB, Input-Output Analysis, Simulation Modeling, Strategic and Critical Materials, Mobilization, Industrial Base Assessments, Investment

15. NUMBER OF PAGES
102

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR